HP Systinet

Software Version: 10.01
Windows and Linux Operating Systems

Administration Guide

Document Release Date: June 2015
Software Release Date: June 2015

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 Notice

© Copyright 2003-2015 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Intel® Xeon® and Intel® Core i7® are registered trademarks of Intel Corporation in the U.S. and other countries.

Microsoft®,Windows®,Windows® XP and Windows 7® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of TheOpenGroup.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Documentation Updates

The title page of this document 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 login page.

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

Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support 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 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 register for an HP Passport ID, go to:

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

To find more information about access levels, go to:

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

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Chapter 1: Administration Overview

Administration in HP Systinet can be broadly divided into the following areas:

- **Managing Content**

  The most important content management concept in Systinet is the use of *Domains*. The administrator can create a domain structure that represents your organizational structure. Each domain represents a working area with users assigned to specific roles within each domain and the content of the domain managed to restrict its visibility and access rights. For more details, see "Domains Overview" on page 15.

  The administrator is also responsible for the day-to-day maintenance of the data content in the Catalog and reports about its status. Systinet provides a set of administration tasks that the administrator can execute manually or schedule to run at set times or periodically to maintain and update this information. For details, see "Administration Task Management" on page 104.

- **Managing Users**

  The management of users is normally delegated to an external user store, such as LDAP, where the management of the people and groups who actually use Systinet should take place. Systinet represents users with *User* artifacts which the particular user or an administrator can manage. Users can create additional *Contact* artifacts to represent external contacts who do not use the product in order to associate them with particular artifacts in the Catalog. For details, see "User Management" on page 24.

  If necessary, the administrator can add and manage additional local groups to organize users into groups that are not represented by the external user store. For details, see "Group Management" on page 30. HP recommends using roles instead of creating local groups.

  An important concept in Systinet is the use of *Roles*. Roles are generic job descriptions that can apply to users and groups in specific domains. The use of roles enables the administrator to manage generic templates for lifecycle processes and security management in the top-level global domain which the resolve to specific users and groups within each working domain. Roles also control user access to functionality in the user interface. For more details, see "Roles Overview" on page 33.

  Administrators within each domain are responsible for assigning users to roles within their domain. For details, see "Domains Overview" on page 15.

- **Managing Security**

  The administrator is responsible for managing and controlling user access to Catalog content. Systinet uses Access Control Lists (ACL) to restrict access based on users, groups, or roles. For details, see "Security and Access Control" on the next page.

- **Managing Global Artifacts**

  Systinet uses domains to divide content into working areas with users assigned to specific roles within the domain. Containing all the working domains is a top-level domain which contains global artifacts which apply across all domains. Systinet restricts access to these artifacts and their management to the top-level administrator. For details, see "Lifecycle Process Management" on page 51 and "Policy Management" on page 43.

- **Configuration and System Management**

  The administrator is responsible for the configuration of each deployment of Systinet.

  The Administration tab provides access to certain aspects of the configuration which can be managed while Systinet is running. For details, see "Configuration Management" on page 76.

  The administrator can customize the user interface. For details, see "UI Customization" on page 109.

  Certain aspects of the configuration are not accessible during runtime. Systinet provides a set of command-line tools and applications that enable this configuration. For details, see "Administration Utilities" on page 252.

Security and Access Control

Most organizations restrict access to resources by user and group permissions. Systinet extends this type of security by enabling the use of domain and role-based access rights.

Systinet uses Access Control Lists (ACL) to define who can access particular resources and their permissions. Each ACL consists of a set of Access Control Elements (ACE) which define the following for a resource or collection of resources:

- **User Identification**

  The user identification as a specified user, a group of users, or a role that resolves to users and groups in the domain that the artifact belongs to.

- **Granted Permission**

  One of the following:

  - **Read Permission**

    Access to read the data and metadata of an artifact or resource, or a collection of artifacts.

  - **Write Permission**

    Access to modify the data and metadata of an artifact or resource, or to create new artifacts, resources, and sub-collections, and update the metadata of a collection of artifacts. Users assigned as the owner of an artifact and administrators always have write permission.

ACLs apply in the following use cases:

- **Artifact Creation Rights**

  The administrator can define which roles can create artifact types within a domain. Within the domain, the users in the allowed roles can access the artifact creation pages for the specified artifact types. The default creation rights are cumulative, so default rights given in the top-level domain apply in all other domains, and rights given to a group or role also apply in addition to rights given to each user in the group or role. For details, see "How to Manage Default Access Rights" on page 20.

- **Governed Artifact Access Rights**

  The access rights for artifacts in governance are determined by the lifecycle process applicable to the artifact. The administrator can assign rights and permissions to particular roles for each stage of a lifecycle process. Within a domain, these roles resolve to the assigned users and groups who have the specified access to the artifact at that lifecycle stage. For details, see "How to Define Permissions" on page 60.

- **Ungoverned Default Artifact Access Rights**

  In the cases where artifacts are not governed, the administrator can define which roles can read or write particular artifact types within a domain. Within the domain, the users in the allowed roles can access the artifact edit pages for the specified artifact types. In addition, you can extend this default access control functionality using particular values of categorization properties. For example, this enables you to define different access rights for services categorized as application services and for services categorized as infrastructure services. The default access rights are cumulative, so default rights given in the top-level domain apply in all other domains, and rights given to a group also apply to all the users of the group. For details, see "How to Manage Default Access Rights" on page 20.

- **Specific Artifact Access Rights**

  The owner, maintainers, and administrators of artifacts can set read or write access rights to artifacts for users, groups, or roles individually or in bulk. For details, see "How to Edit Access Rights" in the *User Guide*.

  Caution: HP recommends only setting specific artifact access rights for ungoverned artifacts.

Chapter 2: Domain Management

Systinet enables the administrator to a create domain structure that reflects the organization of your business. Each artifact in the Catalog belongs specifically to one domain and the administrator can assign users specific roles in specific domains. This enables you to compartmentalize your Catalog and restrict the access and visibility of data to only the users who need it. For more details about the concept of domains, see "Domains Overview" on the next page.

Access domain management from the Administration tab. In the Administration menu, click **Domains** to view the list of all domains. Click a domain name to view its details.

Responsibility for domain management is divided into the following parts:

- **Managing the Domain Structure**

  Users with the Administrator role have responsibility for the overall domain structure. Only the Administrator can create and delete domains. For details, see "How to Create and Delete Domains" on page 18.

- **Managing a Domain**

  The Administrator can assign a separate administrator for specific domains. Users with the administrator role within a domain are responsible for the following:

  - Editing the domain.
  - Setting the default role for the domain. For details, see "How to Manage User Roles in Domains" on page 18.
  - Assigning users to roles in the domain. For details, see "How to Manage User Roles in Domains" on page 18.
  - Managing default permissions in the domain. For details, see "How to Manage Default Access Rights" on page 20.
  - Managing product integration in the domain. For details, see "How to Manage Integrated Products" on page 67.

- **Setting Default Domains**

  When users sign in they access their default domain. The administrator can set default domains for users and groups and individual users can set their own default domain. For details, see "How to Set Default Domains for Users" on page 26, "How to Set Default Domains for Groups" on page 31, and "How to Switch Domains" in the *User Guide*.

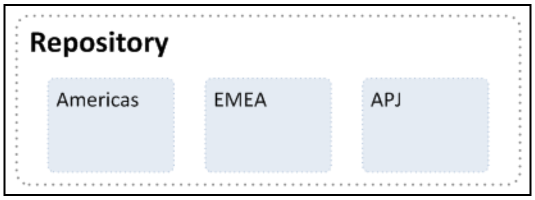
  > **Note:** By default, Systinet contains a default domain which users who are not assigned to a specific working domain sign-in to. To change which domain is the default, change the setting

of the `platform.catalog.defaultUiAdapter.defaultDomain` property to the Domain ID of the required domain. For details, see "How to Manage System Settings" on page 81. The administrator can also disable default domain sign-in to prevent users who are not assigned to a specific working domain from signing-in. For details, see "Disabling Default Domain Sign-In" on page 228.

Domains Overview

Domains provide a logical separation of data within the Catalog. Each domain can represent a discrete working area for an individual department or organizational unit. This separation allows users to focus on the data that is most relevant to them and enables data to be structured by working area.

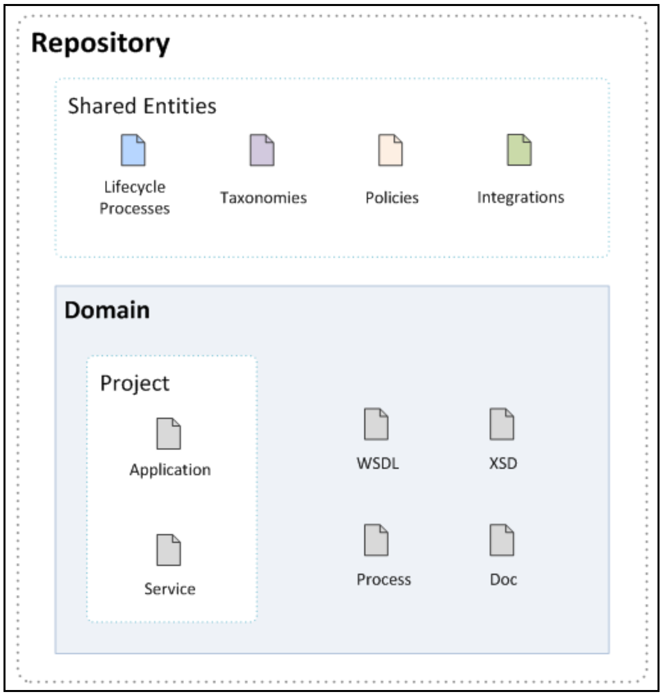
In this release, Systinet provides support for a single layer of domains within a global top-level repository domain. For example, a domain structure representing organizational regions, Americas, EMEA, and APJ.



After installation, Systinet consists of the top-level repository domain and a default domain. The default domain represents a default working area for all users until the administrator creates additional working area domains and assigns users to them.

Note: The administrator can disable default domain sign-in, so that users must be assigned to a specific domain before they can sign-in. For details, see "Disabling Default Domain Sign-In" on page 228.

The top-level domain is a special domain containing system and global data, such as lifecycle processes, policies, and taxonomies, which apply across all domains, and each working domain contains the specific data relevant to users of that domain.



The exact separation of data between the top-level and working domains is as follows:

Top-Level Domain (Global Configuration)

- Lifecycle Processes
- Taxonomies
- Policies
- Roles Definition
- BSM servers
- System settings (including SDM and UI customizations)

Working Domains

- Artifacts (for example, Services, Applications, WSDLs, and Documents) that belong to the domain
- User Role Assignments (for example, Joe is an architect in the EMEA domain)
- Default Settings (for example, a default server folder)
- UDDI Registry / ALM Servers

Working domains inherit all settings applied in the top-level repository domain. For example, access rights, roles, and lifecycle processes set in the top-level domain apply in all domains.

This domain structure creates a logical separation, not only between departments or organizational units, but also between global functions and working area domain functions.

Users in Systinet perform specific functionality based on the roles assigned to them and the user interface restricts their access to functionality and artifacts based in these roles.

The user roles are split into the following user types:

- **Top-Level Repository Administrators**

  Global administration with responsibility for the following functional areas:

  - Domain Management for all domains. For details, see "Domain Management" on page 14.
  - Lifecycle process Administration. For details, see "Lifecycle Process Management" on page 51.
  - Integrated Products Server administration for BSM. For details, see "Product Integration Management" on page 67.
  - User and Group Management. For details, see "User Management" on page 24 and "Group Management" on page 30.
  - Role Administration. For details, see "Role Management" on page 33.
  - Server Configuration Management. For details, see "Configuration Management" on page 76.
  - Policy Administration. For details, see "Policy Management" on page 43.
  - The Administrator can also access all the functionality of Domain Administrators.

- **Domain Administrators**

  Users assigned to the administrator role in a specific domain with responsibility for the following functional areas:

  - Domain Management for their domain. For details, see "Domain Management" on page 14.
  - Management of administrative tasks within their domain. For details, see "Administration Task Management" on page 104.
  - Integrated Products Server administration for UDDI Registries and ALM. For details, see "Product Integration Management" on page 67.

- **Domain Users**

  Users assigned to a specific role within a domain with specific functionality associated with that role. The same user can access different domains in different roles.

This separation of functions and roles is described in more detail in "Roles Overview" on page 33.

Each artifact belongs to exactly one domain. This domain is set to the current domain when a user creates an artifact. Typically, the domain does not change during the artifact lifecycle, but if required it is possible to transfer single artifact or multiple artifacts from one domain to another one using the Change Domain operation. For details, see "How to Edit Artifact Domains" in the *User Guide*.

By default, artifacts are only visible in the owning domain but they can be explicitly shared for all users across all domains using the Share operation. For details, see "How to Share Artifacts" in the *User Guide*. Typically, this operation applies to artifacts entering production and associated with a lifecycle process. For details, see "How to Define Automatic Actions" on page 59.

How to Create and Delete Domains

The Administrator has responsibility for creating and deleting domains within the Catalog.

Access domain management functionality in the Administration tab. In the Administration menu, click **Domains** to open the Domains page. The Domains page provides Add and Delete Domain functionality for Administrators.

To Add Domains:

1. In the Domains page, click **Add Domain** to open the Create Domain page.

2. In the Create Domain page, set a name and description for the domain.

3. *Optional:* Set a Default Domain Location to be the default location for attached data content. For more details, see "Server Folders" in the *User Guide*.

4. Click **Save** to create the new domain.

All working domains exist in a single layer as sub-domains of the top-level global domain. Administrators of the top-level domain have administrative rights in all domains and can assign users and groups to be administrators of specific domains.

To Delete Domains:

1. In the Domains page, select the domains to delete.

2. Click **Delete**, and confirm your choice to delete the selected domains.

> **Note:** You cannot delete a domain if it contains any artifacts. To move artifacts to another domain, see "How to Edit Artifact Domains" in the *User Guide*.

How to Manage User Roles in Domains

The administrator of a domain is responsible for assigning users to roles in the domain. For more details and domains and roles, see "Domains Overview" on page 15 and "Roles Overview" on page 33.

Managing roles in a domain consists of setting a default role and assigning users and groups to specific roles in the domain.

Note: If the administrator assigns a role to a user or a group in the top-level domain, the role assignment applies to all domains.

To Set a Default Role for a Domain:

1. In the Domain detail page select the Overview tab.
2. In the Default Role section, click **(Change)** to open the Choose Role dialog box.
3. Select the role from the list. Optionally, use the filter to find a particular role.
4. Click **Select** to set the selected role as the default role for the domain.

Systinet assigns the selected role to any user who signs into the domain who does not have a role assignment for the domain.

To Assign Users and Groups to Roles for a Domain:

1. In the Domain detail page select the Roles tab.
2. Select the role that you want to add users or groups to.
3. Click **Add Member** to open the Add Member dialog box.
4. Do one of the following:
   - Select the Users tab, and select the users to add. Optionally, use the filter to locate a particular user.
   - Select the Groups tab, and select the groups to add. Optionally, use the filter to locate a particular group.
5. Click **Select** to add the selected users and groups to the selected role in the domain.

To Assign Roles to Users and Groups for a Domain:

1. In the Domain detail page select the Members tab.
2. Click the Edit icon in the Roles column for the user or group you want to assign roles to. The Change Roles dialog box opens.
3. Select the roles that you want to add to the user or group. Optionally use the input dialog to search for a specific role.
4. Click **OK** to add the roles to user or group in the domain.

Once users or groups are assigned to roles within a domain, the administrator can extend their role memberships within specific domains. For details, see "How to Assign Users to Groups and Roles" on page 25 and "How to Assign Groups to Roles" on page 31.

How to Manage Default Access Rights

The administrator of a domain is responsible for defining who is able to create artifacts within the domain they manage. They can also define who has read and write access to artifact types. For more details about domains and security, see "Domains Overview" on page 15 and "Security and Access Control" on page 12.

To access default access rights, in the Administration tab->Administration menu, click **Domains** to open the Domains browse page. Select the domain that you administrate to open its details page and select the **Default Access Rights** tab.

The Default Access Rights tab displays the current Artifact Creation and Read / Write Access permissions for the domain.

> **Note:** Domains inherit all default access rules from the top-level global domain and displays them as non-editable rules for the domain. Any permission rules set within the scope of the domain are additive to those global rules. The administrator of the top-level domain can edit these global rules from the detail page of the top-level domain.

To Manage Artifact Creation Rights:

1. In the Administration tab->Administration menu, click **Domains** to open the Domains browse view.

2. Click the name of the domain you want to set creation rights for to open its details page.

3. Select the **Default Access Rights** tab to view the current permissions for the domain.

4. In the Artifact Creation table, do one of the following:

   - **To Add Artifact Creation Rules:**

     i. Click **Add Rule** to open the Add Artifact Creation Rule page.

     ii. Select the artifact type to add creation rights for from the **To Artifact Type** drop-down list.

     iii. In the Granted To table, click **Add Member** to open the Add Member dialog box.

     iv. Click **Browse Address Book** and select the roles to add creation rights for the selected artifact type. Optionally, use the dialog search input to find a particular role.

     v. Click **Select** to add the selected roles to the Granted To table.

     vi. Click **Save** to add the rule for the selected artifact type to the Artifact Creation table.

   - **To Edit Artifact Creation Rules:**

     i. Click the **Edit** link for the rule to open the Edit Artifact Creation Rule page.

ii. Use the Granted To table to add and remove roles from the rule.

iii. Click **Save** to apply the changes to the rule to the Artifact Creation table.

- **To Remove Artifact Creation Rules:**
  - Select the rules to remove, click **Remove Selected**, and confirm your decision.

To Manage Default Read / Write Access:

1. In the Administration tab->Administration menu, click **Domains** to open the Domains browse view.

2. Click the name of the domain you want to set creation rights for to open its details page.

3. Select the **Default Access Rights** tab to view the current permissions for the domain.

4. In the Read / Write Access table, do one of the following:

   - **To Add Read / Write Access Rules:**

     i. Click **Add Rule** to open the Add Read / Write Access Rule page.

     ii. Select to add **Read** only or **Read / Write** access.

     iii. Select the artifact type to add read / write access for from the **To Artifact Type** drop-down list.

     iv. In the Granted To table, click **Add Member** to open the Add Member dialog box.

     v. Select the roles, users, or groups to add read / write access for the selected artifact type. Optionally, use the dialog search input to find a particular role, user, or group.

     vi. Click **Select** to add the selected roles, users, and groups to the Granted To table.

     vii. *Optional:* Use the Condition table to add conditions to the access rule. These conditions enable you to further restrict access rights based on artifact type specific rules.

     viii. Click **Save** to add the rule for the selected artifact type to the Read / Write Access table.

   - **To Edit Read / Write Access Rules:**

     i. Click the **Edit** link for the rule you want to edit to open the Edit Read / Write Access Rule page.

     ii. Select **Read** only or **Read / Write** access.

     iii. Use the Granted To table to add and remove roles, users, and groups from the rule.

     iv. *Optional:* Use the Condition table to edit conditions for the access rule. These conditions enable you to further restrict access rights based on artifact type specific rules.

v. Click **Save** to apply the changes to the rule to the Read / Write Access table.

- **To Remove Read / Write Access Rules:**
  - Select the rules to remove, click **Remove Selected**, and confirm your decision.

Note: The default read and write permissions only apply to artifacts that are not governed by a lifecycle process or where the initial stage of the governing lifecycle process does not define access rights. For more details about access rights in lifecycle, see "Permissions" in the *User Guide*.

Caution: Do not remove write permission from the Contacts artifact type for the `system#registered` group. This permission is required for the registration of new users.

How to Export Domain Content

The administrator can export the content of an entire domain. Using this functionality for the top-level domain enables you to export the entire content of the Catalog.

Caution: The export / import functionality of the UI is not compatible with the command-line export / import tools available to the administrator. For more details about command-line export / import, see "Export Tool" on page 253 and "Import Tool" on page 256.

To Export Domains:

1. In the Domain Details page Overview tab, click the **Export** context action to open the Export dialog box.

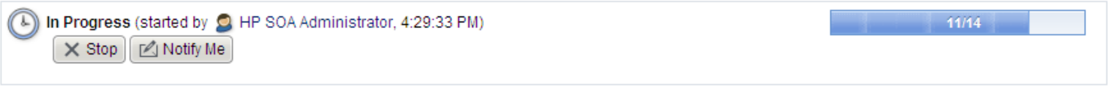
2. *Optional:* Change the name of the archive.

3. *Optional:* Expand **Advanced Options**, and select from the following options:

Advanced Option	Description
Data	Select to export all artifacts that are assigned to the domain.
System Settings	Select which domain settings to export.

4. Click **Export** to create a ZIP archive containing the domain artifacts.

This process executes as a bulk operation. An information bar opens informing you that the operation is in progress with a progress bar with options to **Stop** the operation or to **Notify Me** when the operation is complete.



In Progress (started by HP SOA Administrator, 4:29:33 PM)

The operation executes asynchronously, so you can navigate away and perform other tasks while the operation completes. For more details, see "How to Use Bulk Operations" in the *User Guide*.

5. When the export archive is complete you are prompted for a download location. Alternatively, open the export report and click **Download Content** to save the archive.

The archive is available for import using the Import Repository Archive functionality. For details, see "How to Export and Import Artifacts" in the *User Guide*.

Chapter 3: User Management

Systinet delegates user management to LDAP or an application server user store. Users are represented in Systinet by user artifacts for artifact ownership, notification, and contact purposes.

User artifacts represent users in the user store and contact artifacts represent external contacts. A contact does not have a corresponding LDAP or application user store account and cannot sign in.

When users first sign in, they are authenticated against the external user store and Systinet creates a user artifact based on their external account.

Note: Systinet also checks new users against existing contacts. If a matching contact artifact exists, the login name is attached and the contact becomes a user artifact.

Responsibility for user management is divided into the following parts:

- Each user can manage their own user artifact. For details, see "How to Manage Your Profile" in the *User Guide*.

- Users with appropriate permissions can create new contacts in the Catalog tab. For details, see "How to Create Artifacts" and "Create Contact Page" in the *User Guide*.

  Caution: Only create contacts for people who need to be represented in the Catalog but who do not need to use Systinet. Use LDAP or your application server user store to create users who use Systinet.

- The Administrator is responsible for the following aspects of user management:

  - "How to Disable and Enable Users" on the next page
  - "How to Assign Users to Groups and Roles" on the next page
  - "How to Set Default Domains for Users" on page 26
  - "How to Set New Artifact Ownership" on page 27
  - "How to Import Users from LDAP" on page 28
  - "How to Synchronize Profiles with LDAP" on page 28
  - "Group Management" on page 30
  - "How to Manage Roles" on page 40

How to Disable and Enable Users

The administrator can block access to Systinet by disabling users.

To Disable Multiple Users:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Select the users to disable.
3. Click **Disable** and confirm your selection to block the selected users from accessing Systinet.

To Disable or Enable a Single User:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Click the name of the User you want to Disable or Enable to open their User details page.
3. In the User details page Overview tab, click the **Disable** or **Enable** context action.

How to Assign Users to Groups and Roles

The administrator can assign a user to be a member of multiple groups or roles within specific domains.

To Assign Users to Groups:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Click the name of the User to open its detail page.
3. Select the **Groups and Roles** tab to view the group details for the user.
4. In the Groups table, do one of the following:
   - **To remove the users from groups:**
     i. Select the groups to remove the user from.
     ii. Click Remove and confirm your decision.
   - **To add the users to groups:**
     i. Click **Add to Group** to open the Add Groups to User dialog box.
     ii. Select the groups to add the user to.
     iii. Click **Select** to add the user to the selected groups.

The administrator can also manage the membership of groups from the group perspective. For details, see "How to Manage Group Membership" on page 30.

To Assign Users to Roles:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Click the name of the User to open its detail page.
3. Select the **Groups and Roles** tab to view the role details for the user.
4. In the Roles by Domain section, click **Change** for the required domain to open the Change Role Membership dialog box.
5. Select the additional roles to assign to the user for the domain and click **Set** to apply the changes.

Note: This functionality is only available if the user is already assigned to roles in a domain and can only be used to extend the roles set there. For details, see "How to Manage User Roles in Domains" on page 18.

How to Set Default Domains for Users

The administrator can set the default domain that new users sign in to.

To Set the Default Domain for a Single User:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Click a user name to open their details page.
3. In the Overview tab Access Security section, click the Default Domain **Change** link to open the Set Default Domain dialog box.
4. Select a default domain and click **Select**.

When the user signs in for the first time they access the selected domain.

To Set the Default Domain for Multiple Users:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Expand Manage Members, and select **Set Default Domain** to open the Set Default Domain dialog box.
3. Select a domain from the list and click **Select** to set the domain as the default for new users.

When the selected users sign in for the first time they access the selected domain.

Default domains can also be set for groups and individual users can set their own default domain. For details, see "How to Set Default Domains for Groups" on page 31 and "How to Switch Domains" in the *User Guide*.

How to Set New Artifact Ownership

By default, users who create artifacts become the *owner* of the artifact. The owner of an artifact has default read / write permission for the new artifact.

The administrator can configure new artifact ownership for users so that instead of the artifact ownership being assigned to the user, it is instead assigned to a specified role or group.

To Set New Artifact Ownership for a Single User:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Click a user name to open their details page.
3. In the Overview tab Access Security section, click the New Artifact Ownership **Change**link to open the Select New Artifacts Ownership dialog box.
4. Click **Change** to select a new owner.
5. Input a role or group search term or click **Browse Address Book** and select from the list of roles or groups.

   Note: The users must be members of the selected group or assigned to the selected role in at least one domain.
6. Click **Select** to apply the new artifact ownership assignment to the selected users.

To Set New Artifact Ownership for Multiple Users:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.
2. Select the users to apply new artifact ownership assignment to.
3. Expand Manage Members, and select **Set New Artifacts Ownership** to open the Set New Artifacts Ownership dialog box.
4. Click **Change** to open the dialog box.
5. Input a role or group search term or click **Browse Address Book** and select from the list of roles or groups.

   > **Note:** The users must be members of the selected group or assigned to the selected role in at least one domain.
6. Click **Select** to apply the new artifact ownership assignment to the selected users.

When the selected users create artifacts, Systinet assigns ownership to the specified role or group.

How to Import Users from LDAP

If Systinet is synchronized with an LDAP user store you can create user artifacts based on corresponding LDAP accounts.

To Import LDAP Accounts:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.

2. Click **Import** to open the Select Import Users dialog box.

3. Do one of the following:

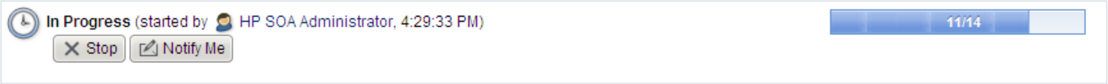
   - Use the Search input and click **Search** to populate the Users list with users in the LDAP user store matching the search term.

     > **Tip:** Select a user from the *as-you-type* drop-down to immediately create a matching user artifact.

   - Click **Browse Address Book** to populate the Users list with all the users in the LDAP user store.

4. Select users from the Users or Groups list, and click **Select** to create matching user artifacts.

   This process executes as a bulk operation. An information bar opens informing you that the operation is in progress with a progress bar with options to **Stop** the operation or to **Notify Me** when the operation is complete.

   In Progress (started by HP SOA Administrator, 4:29:33 PM)

   The operation executes asynchronously, so you can navigate away and perform other tasks while the operation completes. For more details, see "How to Use Bulk Operations" in the *User Guide*.

> **Note:** Import uses the LDAP cache so changes in LDAP may not be immediately visible in Systinet.

How to Synchronize Profiles with LDAP

If Systinet is synchronized with an LDAP user store you can update user artifacts with the latest versions of the corresponding LDAP accounts.

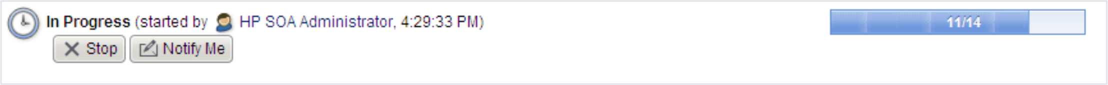
To Synchronize Profiles with LDAP Accounts:

1. In the Administration tab->Administration menu, click **Users** to open the Users browse page.

2. In the Users browse page, select the users you want to synchronize.

3. Click **Synchronize** and confirm your decision.

   This process executes as a bulk operation. An information bar opens informing you that the operation is in progress with a progress bar with options to **Stop** the operation or to **Notify Me** when the operation is complete.

   In Progress (started by HP SOA Administrator, 4:29:33 PM)

   The operation executes asynchronously, so you can navigate away and perform other tasks while the operation completes. For more details, see "How to Use Bulk Operations" in the *User Guide*.

Note: Synchronize uses the LDAP cache so changes in LDAP may not be immediately visible in Systinet.

Chapter 4: Group Management

In addition to any groups managed by the external user store, the administrator can manage local groups in Systinet.

> HP Software recommend using roles instead of local groups.

Access group management from the Administration tab. In the Administration menu, click **Groups** to view the list of all groups. Click a group name to view its details.

To Create Groups:

- In the Groups page, click **Create Group** to open the New Group page.
- Enter a name and description, and optionally add group members as described in "How to Manage Group Membership" below.

The administrator is also responsible for the following aspects of group management:

- "How to Manage Group Membership" below
- "How to Assign Groups to Roles" on the next page
- "How to Set Default Domains for Groups" on the next page
- "How to Retire and Delete Groups" on page 32

How to Manage Group Membership

The administrator is responsible for managing the membership of local groups.

To Manage Group Membership:

1. In the Administration tab->Administration menu, click **Groups** to open the Groups browse page.
2. Click the name of the group to open its details page.
3. Click **Edit** to open the Edit Group page.
4. In the Members table, do one of the following:
   - **To remove users from the group:**
     1. Select the users to remove from the group.
     2. Click **Remove** and confirm your decision.

- **To add users to the group:**
    1. Click **Add Member** to open the Add Users to Group dialog box.
    2. Select the users to add to the group.
    3. Click **Select** to add the selected users to the group.

5. Click **Save** to apply your group membership changes.

The administrator can also manage the membership of groups from the user perspective. For details, see "How to Assign Users to Groups and Roles" on page 25.

How to Assign Groups to Roles

The administrator can assign a group to be a member roles within specific domains.

To Assign Groups to Roles:

1. In the Administration tab->Administration menu, click **Groups** to open the Groups browse page.
2. Click the name of the Group to open its detail page.
3. In the Roles by Domain section, click **Change** for the required domain to open the Change Role Membership dialog box.
4. Select the additional roles to assign to the group for the domain and click **Set** to apply the changes.

Note: This functionality is only available if the group is already assigned to roles in a domain and can only be used to extend the roles set there. For details, see "How to Manage User Roles in Domains" on page 18.

How to Set Default Domains for Groups

The administrator can set the default domain that members of groups sign in to.

To Set the Default Domain for Groups:

1. In the Administration tab->Administration menu, click **Groups** to open the Groups browse page.
2. Click the name of the group to open its details page.
3. Click **Set Default Domain** to open the Set Default Domain dialog box.
4. Select a domain from the list and click **Select** to set the domain as the default for the group.

When members of the selected group sign in they access the selected domain.

Default domains can also be set for users and individual users can set their own default domain. For details, see "How to Set Default Domains for Users" on page 26 and "How to Switch Domains" in the *User Guide*.

How to Retire and Delete Groups

The Administrator can retire and then delete internal groups.

> **Note:** These processes do not retire or delete the users who are members of these groups.

To Retire Multiple Groups:

- In the Groups browse page, select the groups to retire, click **Retire** and confirm your decision.

To Retire a Single Group:

1. In the Group details page, click the **Retire Group** context action to open the Retire Group dialog box.

2. Click **(Change)** to select a new owner for any artifacts owned by members of the group.

To Delete Multiple Groups:

- In the Groups browse page, select the groups to delete, click **Delete** and confirm your decision.

To Delete a Single Group:

- In the Group details page, click the **Delete Group** context action.

  > **Note:** A group must be retired before you can delete it.

Chapter 5: Role Management

Systinet enables the administrator to assign users to specific roles related to their job functions. These roles restrict their access to artifacts, and limit their functionality only to that appropriate to their role. For more details about the concept of roles, see "Roles Overview" below.

Access role management from the Administration tab. In the Administration menu, click **Roles** to view the list of all roles. Click a role name to view its details.

Responsibility for role management is divided into the following parts:

- **Managing Roles**

  Users with the Administrator role have responsibility for the roles available in Systinet. Only the Administrator of the top-level domain can create and delete roles. For details, see "How to Manage Roles" on page 40.

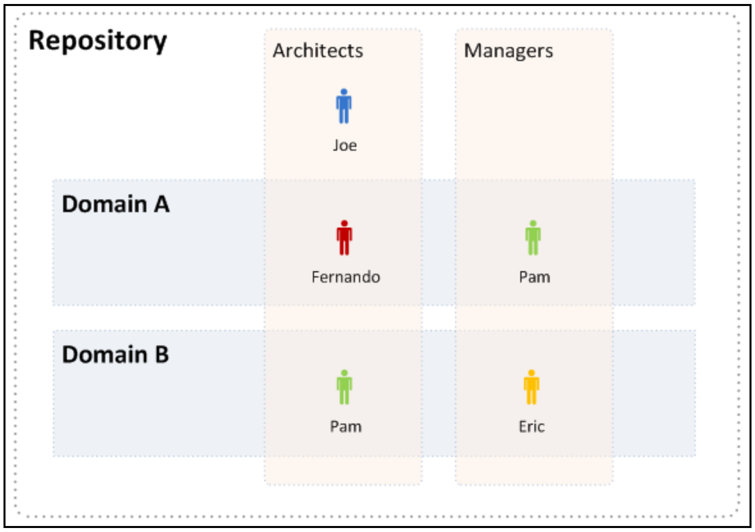
- **Assigning Users to Roles**

  Users with the administrator role within a domain are responsible assigning users to specific roles within their domain and setting a default role for new users of the domain. For details, see "How to Manage User Roles in Domains" on page 18.

Roles Overview

Systinet offers functionality across the entire service development lifecycle. In most organizations, these functions are performed by many individuals and teams with specific permissions. Systinet uses *Roles* to enable you to define and assign these permissions, and use these assignments to focus each user or group on specific functionality and tasks and restrict their access to artifacts appropriate to their role.

The administrator defines roles in the top-level domain, but user assignment to roles can be global or to different roles in different domains. For example, in the following diagram, Joe is assigned a global architect role in the top-level repository domain, Fernando is assigned the architect role in Domain A, Pam is assigned the manager role in Domain A and the architect role in Domain B, and Eric is assigned to the manager role in Domain B.



These assignments mean that in Domain A, Joe and Fernando access functionality and artifacts relevant to the architect role, whereas Pam accesses functionality relevant to the manager role.

Pam has a different role in Domain B, so along with Joe, accesses architect functionality, whereas Eric accesses functionality relevant to the manager role.

To assign users or groups to roles, see "How to Manage User Roles in Domains" on page 18. For more details about domains, see "Domains Overview" on page 15.

The following topics describe in more detail how Systinet uses roles:

- "Roles in the User Interface" on the next page

  The UI uses roles to restrict the availability of functionality to users in appropriate roles.

- "Roles in Lifecycle" on page 36

  You can create Lifecycle templates with specified tasks and actions assigned to specific roles.

- "Security and Access Control" on page 12

  Systinet restricts access to artifact types using ACLs which can use roles as well as users and groups.

The default roles and their assigned functionality are described in the following topics:

- "Business Partner Role" on page 37
- "Business Analyst Role" on page 38
- "Service Provider Role" on page 38
- "Administrator Role" on page 39

The administrator can extend the default roles by adding additional customized roles. For details, see "How to Manage Roles" on page 40.

Systinet also includes a special role, Sharing Principal, specifically associated with sharing artifacts. By default, this role is associated with the `system#registered` group which represents all users who access Systinet. For more details, see "How to Change the Sharing Principal" on page 41 .

Roles in the User Interface

Systinet restricts access to UI functionality according to your role.

- **Business Partner**

  The Catalog tab provides search and browse functionality for consumable artifacts and enables users in the Business Partner role to request contracts. For more details, see "Business Partner Role" on page 37.

  The Navigator tab provides a visual representation of the relationships between artifacts. It provides various layouts and role-based filters to enable you to visualize the content of the repository relevant to your role. For more details, see "Navigator Overview" in the *User Guide*.

- **Business Analyst**

  The Catalog tab for Business Analysts extends Business Partner functionality. It enables users in the Business Analyst role to search and browse service-related artifacts, to interact with lifecycle, and to create service-related artifacts. For more details, see "Business Analyst Role" on page 38.

  The Navigator tab provides a visual representation of the relationships between artifacts. It provides various layouts and role-based filters to enable you to visualize the content of the repository relevant to your role. For more details, see "Navigator Overview" in the *User Guide*.

- **Service Provider**

  The Catalog tab for Service Providers extends Business Analyst functionality. It enables users in the Service Provider role to create, develop, and manage service-related artifacts. For more details, see "Service Provider Role" on page 38.

  The Navigator tab provides a visual representation of the relationships between artifacts. It provides various layouts and role-based filters to enable you to visualize the content of the repository relevant to your role. For more details, see "Navigator Overview" in the *User Guide*.

  The Reports tab provides access to view and create reports about Catalog content. For more details, see "Reporting Overview" in the *User Guide*.

- **Administrator**

  In addition to all the tabs and functionality accessible by users in the Service Provider role, administrators access an Administration tab to enable them to manage users, groups, roles, domains, and other system artifacts. For more details, see "Administration Overview" on page 11.

Roles in Lifecycle

Lifecycle management makes use of roles to enable global lifecycle process management with role-based assignments in a lifecycle process template.

The administrator of the top-level domain manages lifecycle processes and uses roles to define the following:

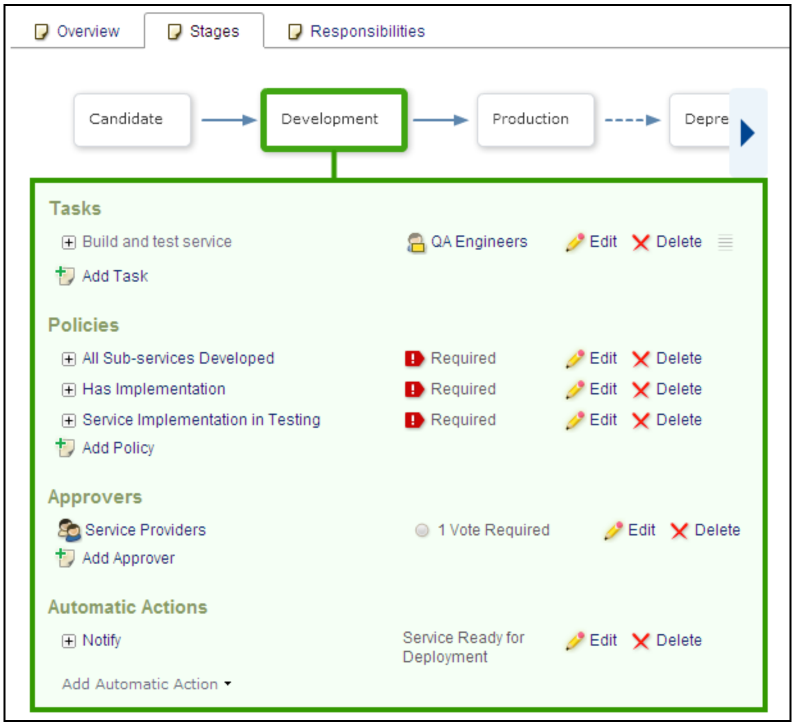
- The user role responsible for approving particular lifecycle stages.
- The user roles with read-only, write access, and ownership permissions for artifacts at particular lifecycle stages.
- The user roles responsible for performing tasks associated with a lifecycle stage.
- The user roles that are automatically notified as a result of specific events within the lifecycle.

For more details about lifecycle governance, see "Lifecycle Overview" in the *User Guide*.

Within specific domains, these role assignments resolve to the users and groups assigned to the role in that domain. For example, consider the development stage of a lifecycle process for services.

The stage might consist of the following role assignments:

- A task to build and test the service assigned to the QA Engineer role.
- A stage approver in the Service Provider role.
- An automatic action to notify the users in the Operations Manager role when a service is approved at the development stage.



Tasks

⊞ Build and test service	QA Engineers	Edit	Delete
Add Task			

Policies

⊞ All Sub-services Developed	Required	Edit	Delete
⊞ Has Implementation	Required	Edit	Delete
⊞ Service Implementation in Testing	Required	Edit	Delete
Add Policy			

Approvers

Service Providers	○ 1 Vote Required	Edit	Delete
Add Approver			

Automatic Actions

⊞ Notify	Service Ready for Deployment	Edit	Delete
Add Automatic Action ▾			

In different domains, different users perform each of these roles. Within domains, Systinet replaces the roles in the template with the specific users and groups assigned to that role in that domain.

In the EMEA domain, this could result in the following specific assignments:

- A member of the EMEA QA Engineers group, assigned to the QA Engineer role, must complete the Build and Test task.

- An EMEA domain user, assigned to the Service Provider role, must approve the development stage for the service.

- When the service is approved, all users in the Operations Manager role in the EMEA domain are notified that the service is ready for deployment.

In the US domain these assignments are to different users and groups performing the same roles.

Business Partner Role

Business Partner is a consumer role, typically an external business-to-business user, group or a company. Typically, service consumers are seeking to reduce time to market and reduce the cost of the development. They build composite applications by searching for existing services and negotiating contracts with service providers.

Users in the Business Partner role perform the following functions:

- Search and Discover consumable service artifacts. For details, see "Search and Browse Overview" in the *User Guide*.
- Visualize the Catalog content. For details, see "Navigator Overview" in the *User Guide*.
- Collaborate with your colleagues and service providers. For details, see "Collaboration Overview" in the *User Guide*.
- Request contracts to consume service artifacts. For details, see "Contracts Overview" in the *User Guide*.

Business Analyst Role

A Business Analyst is a negotiator between the business side of an enterprise and the providers of services to the enterprise. The business analyst understands business problems and opportunities in the context of the requirements and recommends solutions that enable the organization to achieve its goals. Solutions often include a systems development component, but may also consist of process improvement or organizational change.

Users in the Business Analyst role perform the following functions:

- Search and Discover consumable service artifacts. For details, see "Search and Browse Overview" in the *User Guide*.
- Visualize the Catalog content. For details, see "Navigator Overview" in the *User Guide*.
- Collaborate with your colleagues and service providers. For details, see "Collaboration Overview" in the *User Guide*.
- Author new Catalog content. For details see "Authoring Overview" in the *User Guide*.
- Edit existing Catalog content. For details, see "Artifact Management Overview" in the *User Guide*.
- Create new versions of existing Catalog content. For details, see "Versioning Overview" in the *User Guide*.
- Request, approve and revoke contracts to consume service artifacts. For details, see "Contracts Overview" in the *User Guide*.
- Participate in the lifecycle of Catalog content. For details, see "Lifecycle Overview" in the *User Guide*.

Service Provider Role

The *Service Provider* creates, develops, manages, and provides services, applications, and processes. In contrast to Business Partner they are interested in the design-time policies, test cases, functional specifications, services in different stages, and reporting tools.

Users in the Service Provider role perform the following functions:

- Search and Discover consumable service artifacts. For details, see "Search and Browse Overview" in the *User Guide*.
- Visualize the Catalog content. For details, see "Navigator Overview" in the *User Guide*.
- Collaborate with your colleagues and service providers. For details, see "Collaboration Overview" in the *User Guide*.
- Author new Catalog content. For details see "Authoring Overview" in the *User Guide*.
- Edit existing Catalog content. For details, see "Artifact Management Overview" in the *User Guide*.
- Create new versions of existing Catalog content. For details, see "Versioning Overview" in the *User Guide*.
- Request, approve and revoke contracts to consume service artifacts. For details, see "Contracts Overview" in the *User Guide*.
- Participate in the lifecycle of Catalog content. For details, see "Lifecycle Overview" in the *User Guide*.
- View and create reports on Catalog content. For details, see "Reporting Overview" in the *User Guide*.
- Export, import, synchronize Catalog content with integrated products and review the data they return. For details, see "Product Integration Overview" in the *User Guide*.

Administrator Role

Systinet provides an administrator role with responsibility for managing users, groups, roles, and system artifacts. The responsibilities of administrators vary according to the domain that they manage.

- **Top-Level Repository Administrators**

  Global administration with responsibility for the following functional areas:

  - Domain Management for all domains. For details, see "Domain Management" on page 14.
  - Lifecycle Process Administration. For details, see "Lifecycle Process Management" on page 51.
  - Integrated Products Server administration for BSM. For details, see "Product Integration Management" on page 67.
  - User and Group Management. For details, see "User Management" on page 24 and "Group Management" on page 30.
  - Role Administration. For details, see "Role Management" on page 33.

- Server Configuration Management. For details, see "Configuration Management" on page 76.
- Policy Administration. For details, "Policy Management" on page 43.
- The Administrator can also access all the functionality of Domain Administrators.

- **Domain Administrators**

  Users assigned to the administrator role in a specific domain with responsibility for the following functional areas:

  - Domain Management for their domain. For details, see "Domain Management" on page 14.
  - Management of administrative tasks within their domain. For details, see "Administration Task Management" on page 104.
  - Integrated Products Server administration for UDDI Registries and ALM. For details, see "Product Integration Management" on page 67.

How to Manage Roles

The Administrator has responsibility for managing roles in Systinet. For details about the concept of roles, see "Roles Overview" on page 33.

To view the list of roles in Systinet, in the Administration tab->Administration menu, click **Roles** to open the Roles browse page.

The Roles browse page provides the following functionality:

- **Delete**

  Select the roles to delete, and click **Delete**.

  Note: You cannot delete the default roles.

- **Create Role**

  Click **Create Role** to open the "Create Role Page" on the next page.

After you create a role, it is available for user and group assignment, for use in lifecycle processes, and for default access rights.

Click a role name to open its details page, showing its properties and UI Access details.

Click **Edit** to change the details for a role. For system roles, you can only change the UI Access details. For details of the Edit Role page parameters, see "Create Role Page" on the next page.

For details about using roles, see the following topics:

- "How to Manage User Roles in Domains" on page 18
- "How to Create Lifecycle Processes" on page 51
- "How to Edit Access Rights" in the *User Guide*
- "How to Manage Default Access Rights" on page 20

How to Change the Sharing Principal

The Sharing Principal is a special role associated with sharing artifacts to make them visible to more users. The default functionality is to use the `system#registered` group which represents all users who access Systinet.

To Change the Sharing Principal:

1. In the Administration tab->Administration menu, select **Domains** to open the Domains page.
2. Select the Top-Level domain to open the Domains details page.
3. Select the Roles tab to view the roles and their membership for the domain.
4. Select the Sharing Principal role to view its membership.
5. Use **Remove Selected** and **Add Member** to edit the membership of the role.

The selected users and groups see any artifacts that are shared using the functionality described in "How to Share Artifacts" in the *User Guide*.

Create Role Page

The Create and Edit Role pages contain the following parameters:

Parameter	Definition
Name	Name of the Role.
Description	A description including rich text and HTML support. This description displays in the Artifact Details page Overview tab properties area. For details, see *User Guide* "Artifact Details Page Overview Tab" in the . This description is limited by the display length available in the Overview tab. If you need a longer description or an artifact specification edit the Specification property. For details, see "How to Edit Artifact Specifications" in the *User Guide* .
Sub-Roles	Select from the set of existing roles to inherit their functionality and artifact access for the new role.

Parameter	Definition
Additional UI Access	Select whether users in the new role can access additional specified tabs.
Allows Login	Allow users in the new role to access the UI.

Chapter 6: Policy Management

Systinet enables you to validate your Catalog content against published policies to ensure its consistency and conformance to your business policy.

Policy management and validation uses the following artifact types:

- **Technical Policy**

  A technical policy consists of a set of assertions and references to other technical policies, and serves as the central point of reference for validation. References to other technical policies enable you to collect a set of technical policies together into a larger policy enabling you to validate them collectively.

- **Assertion**

  An assertion is a validation check for a single piece of data which can either pass or fail. For example, an assertion can check the following attributes of an artifact:

  - **Property Values**

    An assertion can check that a particular property has a value and what the value is. Typical examples include verifying that keywords are set when an artifact is created and verifying that an artifact is consumable before it enters the production lifecycle stage.

  - **Related Artifacts**

    An assertion can check whether a particular artifact type is associated with the artifact being verified and check property values of the related artifacts. A typical example is to verify that a service has an attached document which is categorized as a business specification document before the service is approved at the candidate lifecycle stage.

  - **Data Content**

    Some artifact types are expected to contain attached content. An assertion can verify that the attachment exists.

  - **Artifact State**

    An assertion can check various status attributes of an artifact, For example, its lifecycle status or its compliance status.

  - **Metrics from Integrated Products**

    An assertion can check values generated by integrated products. For example, whether there are any open defects in *HP Application Lifecycle Manager* (ALM).

  Create and edit assertions using Assertion Editor. For details, see the *Assertion Editor Guide*.

Systinet enables you to use technical policies in the following ways:

- **Lifecycle Stage Validation**

  Lifecycle stages in a lifecycle process can have a set of policies associated with them that serve as validation checks for artifacts at that lifecycle stage. Policies can be optional or required and can also validate that tasks associated with the stage are complete. These policies are automatically validated when a user requests stage approval or can be manually validated by users. The collective status of these policies gives a Compliance Status for an artifact which measures the current percentage of required policies which pass validation. The administrator assigns policies to lifecycle stages during the creation of lifecycle processes. For details about setting up and performing lifecycle validation, see "How to Define Policies" on page 56 and "How to Review/Validate Policies" in the *User Guide*.

  Systinet includes a default Lifecycle Validation Task, scheduled to run once a day, which automatically validates all artifacts in governance against the policies that apply to their current lifecycle stage. For details about managing this task, see "Administration Task Management" on page 104.

- **Manual Validation**

  You can use the Policy Report feature of the Reports tab to perform ad-hoc manual validation. You can set up a policy report which validates a selected set of artifacts against a set of selected technical policies. For details, see "Policy Reports" in the *User Guide*.

- **Artifact Form Validation**

  The administrator can configure a set of technical policies that validate an artifact whenever it is created or modified. You can use this type of validation to ensure that new and amended artifacts contain a minimum set of descriptive data such as keywords and impact type, and enforce rules such as the versioning schema. Each artifact type can have its own set of form validation. For details about configuring form validation, see "How to Manage Artifact Form Validation" on page 102.

- **IDE Integration**

  HP Software also provides a set of IDE plugins with integrated policy validation functionality. For details, see *Plugin for Eclipse* and *Plugin for Visual Studio*.

The administrator of the top-level domain is responsible for policy management. For procedural details, see the following topics:

- "How to Manage Technical Policies" below
- "How to Manage Assertions" on the next page

How to Manage Technical Policies

The administrator manages technical policies in the Administration tab.

In the Administration tab->Administration menu, click **Technical Policies** to view the list of policies in the Catalog.

The Technical Policies page provides the following management functions.

- **Create Technical Policy**

  Click **Create Technical Policy** and refer to "Create Technical Policy Page" on page 50 for parameter details.

- **Delete Technical Policies**

  Select the technical policies to delete, and click **Delete**. For more details, see "How to Delete Artifacts" in the *User Guide*.

The **Applicable To** column enables you to filter the policies table according to the artifact type that it is applicable to.

Click the name of a technical policy to view its details.

The technical policy detail page provides access to individual technical policy management tasks. From this page you can perform the following functions:

Add Technical Policy

Collect a set of related policies together under a single technical policy, This enables you to use the parent policy to perform validation against the set of policies. Click **Add Technical Policy** to add the first technical policy, and then the standard table functionality for related artifacts. For details, see "How to Manage Relationships" in the *User Guide*.

Add Assertion

Manage the assertions related to the technical policy using standard table functionality. For details, see "How to Manage Relationships" in the *User Guide*. After selecting an assertion to add you must set its scope within the technical policy. Set the severity, and any parameters defined by the assertion.

> **Note:** Parameters inherit any default values set in the assertion template and any non-optional parameters must be set. For more details, see "Assertion Schema" in the *Reference Guide*.

You can change the severity and parameters for an assertion using the **Edit** link for the assertion in the table of assertions.

In addition the Technical Policy details page provides the standard edit, delete, and export artifact management tasks. For details, see "Artifact Management Overview" in the *User Guide*.

How to Manage Assertions

The administrator manages assertions in the Administration tab.

In the Administration tab->Administration menu, click **Assertions** to view the list of policies in the Catalog.

The Assertions page provides the following management functions.

- **Delete Assertions**

  Select the assertions to delete, and click **Delete**. For more details, see "How to Delete Artifacts" in the *User Guide*.

> **Note:** The creation and editing of assertions is delegated to *Assertion Editor*. For details, see the *Assertion Editor Guide*.

Click the name of a assertion to view its details.

The Technical Details tab displays the code of the assertion.

In addition the Assertion details page provides the standard edit, delete, and export artifact management tasks. For details, see "Artifact Management Overview" in the *User Guide*.

As part of a technical policy, the scope of an assertion is editable. For details, see "How to Manage Technical Policies" on page 44.

Default Technical Policies

Systinet includes a prepackaged set of technical polices.

These technical polices are described in the following tables organised by the type of validation they perform:

Related Artifact Validity: Policies that verify that relevant related artifacts are present for a service, such as documentation

Technical Policy	Definition
Has Business Requirements	Verifies that an artifact has a business requirements document.
Has Project Plan	Verifies that an artifact has a project plan document.
Has Detailed Design	Verifies that a service has a detailed design document.
Has Functional Specification	Verifies that a service has a functional specification document.
Has Installation Instructions	Verifies that a service has an installation instruction document.
Has Technical Specification	Verifies that a service has a technical specification document.
Has User Manual	Verifies that a service has a user manual document.
Has Implementation	Verifies that a service has an implementation.
Has SLO	Verifies that a service has a service level agreement.
Project Has Analysis	Verifies that a project artifact has an analysis document.

Service Property Validity: Policies that verify properties of a service

Technical Policy	Definition
Consumption Is Not Allowed	Verifies that a service is not consumable.
Is Consumable	Verifies that a service is consumable
Contact Person Is Assigned	Verifies that a contact is assigned for a service.
Failure Impact Is Set	Verifies that a failure impact status is set for a service.
Is Described By Keywords	Verifies that keywords describing the service are set.

Lifecycle Process Dependency Validity: Policies for lifecycle processes that verify the lifecycle stage of constituent artifacts governed by other processes

Technical Policy	Definition
All Application Components Deprecated	Verifies that the constituent artifacts of an application are all approved at the Deprecated stage.
All Application Components Developed	Verifies that the constituent artifacts of an application are all approved at the Development stage.
All Application Components Retired	Verifies that the constituent artifacts of an application are all approved at the Retired stage.
All Application Components in Production	Verifies that the constituent artifacts of an application are all approved at the Production stage.
All Process Implementations Deprecated	Verifies that all referenced process implementations are approved at the Deprecated stage.
All Process Implementations Retired	Verifies that all referenced process implementations are approved at the Retired stage.

Lifecycle Process Dependency Validity: Policies for lifecycle processes that verify the lifecycle stage of constituent artifacts governed by other processes, continued

Technical Policy	Definition
All Project Artifacts Developed	Verifies that the constituent artifacts of a project are all approved at the Development stage.
All Project Artifacts in Production	Verifies that the constituent artifacts of a project are all approved at the Production stage.
All Service Implementations Deprecated	Verifies that all referenced service implementations are approved at the Deprecated stage.
All Service Implementations Retired	Verifies that all referenced service implementations are approved at the Retired stage.
All Sub-Processes Developed	Verifies that all referenced sub-processes are approved at the Development stage.
All Sub-Processes in Production	Verifies that all referenced sub-processes are approved at the Production stage.
All Sub-Processes Developed	Verifies that all referenced sub-services are approved at the Development stage.
All Sub-Processes in Production	Verifies that all referenced sub-services are approved at the Production stage.
Process Implementation in Production	Verifies that at least one referenced process implementation is approved at the Production stage.
Process Implementation in Testing	Verifies that at least one referenced process implementation is approved at the Testing stage.
Service Implementation in Production	Verifies that at least one referenced service implementation is approved at the Production stage.
Service Implementation in Testing	Verifies that at least one referenced service implementation is approved at the Testing stage.

WSDL and SOAP Message Validity: Policies that verify WSDL and SOAP conformance with standard schema

Technical Policy	Definition
WSDL Validity Policy	Verifies that a WSDL is compliant with the WSDL 1.1 XML Schema, that all references to external WSDL descriptions and schemas are valid, and that the WSDL is compliant with WS-I Basic Profile 1.1.
WS-I Basic Profile Policy	Verifies that a WSDL or SOAP message complies with WS-I Basic Profile 1.1.
XSD Validity Policy	Checks that an XML Schema is valid and that all contained references are resolvable.
SOAP Validity Policy	Verifies that SOAP messages are compliant with SOAP 1.1 XML Schema and WS-I Basic Profile 1.1.

Service Status Validity: Policies that verify the status of services

Technical Policy	Definition
Service In Production Stage	Verifies that a service has contact information, is documented, has an implementation, is described with keywords, and that its associated WSDL documents and XML schema comply with WS-I Basic Profile 1.1 and the "Systinet Best Practices" policy (see below).
Service In Testing Stage	Verifies that a service has an implementation and that associated WSDL documents and XML schemas comply with WS-I Basic Profile 1.1 and the "Systinet Best Practices" policy (see below).
Service Minimum Production Requirements	Verifies the basic requirements of production service; that it has contact information, is documented, has an implementation, and is described with keywords. The associated WSDL documents and XML schemas must comply with WSDL and XML schema specifications and all references to types can be resolved.
Systinet Best Practices	Verifies that WSDL documents XML schemas have correct extensions, that names contain valid characters, that schema use XML schema 2001, namespace prefix "xs", that types and elements are annotated, etc.

Application Lifecycle Management Policies: Policies to verify the status of services in conjunction with HP Application Lifecycle Management (ALM) integration

Technical Policy	Definition
Service Registered in ALM	Verifies that a service is registered in ALM.

Application Lifecycle Management Policies: Policies to verify the status of services in conjunction with HP Application Lifecycle Management (ALM) integration, continued

Technical Policy	Definition
Service Has No Pending Critical Defects	Verifies that a service is registered in ALM and that there are no critical defects outstanding for the service.
Service Has Test Requirements and Tests	Verifies that a service is registered in ALM and that there are test requirements and tests defined for the service in ALM.
Test Requirements Covered	Verifies that a service is registered in ALM and that the test requirements report meets the criteria defined in ALM.

Create Technical Policy Page

The Create and Edit Technical Policy pages contain the following parameters:

Parameter	Definition
Name	Name of the technical policy.
Description	A description including rich text and HTML support. This description displays in the Artifact Details page Overview tab properties area. For details, see *User Guide* "Artifact Details Page Overview Tab" in the . This description is limited by the display length available in the Overview tab. If you need a longer description or an artifact specification edit the Specification property. For details, see "How to Edit Artifact Specifications" in the *User Guide* .
Technical Policies	Use the **Add Technical Policy** functionality of the table to manage referenced technical policies. For details, see "How to Manage Relationships" in the *User Guide*.
Assertions	Use the **Add Assertion** functionality of the table to manage referenced assertions. For details, see "How to Manage Relationships" in the *User Guide*.

Chapter 7: Lifecycle Process Management

Before a Lifecycle Process can be used, it must be first defined and published by a user with Administrator rights.

A Lifecycle Process can also be copied and modified so as to maintain certain values rather than having to be constructed from the beginning.

Lifecycle Processes are defined by clicking **Lifecycle Processes** under the Administration tab and either selecting an existing process for modification or clicking **Create** to build a new one.

In the Lifecycle Process Overview or Stages tab it is possible to edit, publish, clone, delete or export the current process by clicking on the appropriate action.

The following tasks may be required when defining a new, or editing an existing Lifecycle Process:

- "How to Create Lifecycle Processes" below
- "How to Define Stages" on page 53
- "How to Define Tasks" on page 56
- "How to Define Transitions" on page 55
- "How to Define Approvers" on page 57
- "How to Define Policies" on page 56
- "How to Define Permissions" on page 60
- "How to Define Automatic Actions" on page 59
- "How to Publish a Process" on page 61
- "How to Export Lifecycle Processes" on page 62

"Lifecycle Best Practice" on page 62 describes HP recommended principles for lifecycle governance and describes the default lifecycle processes included with Systinet. For more details, see .

See "Lifecycle Overview" in the User Guide for the user view of Lifecycle Processes.

How to Create Lifecycle Processes

The first step in artifact lifecycle management is the creation of a lifecycle process.

To create a new lifecycle process:

1. In the Administration menu, click **Lifecycle Processes** to open the Lifecycle Processes page, and then click **Create**.
2. In the upper section, enter a **Name** and **Description** for the lifecycle process.
3. Define the lifecycle stages for the lifecycle process as follows:
   a. Select the Root Artifact Type from the drop down list.
   b. Optionally select the Sub-Artifact Type(s) by selecting the check box of the required types
4. If the new process is to be automatically assigned to artifacts, click the check box and choose either to automatically assigned by script or by categories.

   To assign artifacts automatically by script, follow these steps:

   Create an embedded script in Manage Scripts menu, set Execute on = Auto Assignment Lifecycle

   a. Add below content:

   ```
   function isAssignable(rootArtifact)

   {

   // do the calculation here

   // “return true;” if rootArtifact should be automatically assigned to
   lifecycle process

   // “return false;” if rootArtifact shouldn’t be automatically assigned to
   lifecycle process

   }
   ```

   b. Open the lifecycle process to be automatically assigned, then select this script in Automatic Assignment Script

   To assign artifacts automatically by categories, add categories for Categorized as and Not Categorized as to apply taxonomic association rules for the automatic assignment of the lifecycle process to artifacts based on their categorizations

   a. **In Categories**: Artifacts must contain these categories to be automatically assigned to the lifecycle process.
   b. **Not In Categories**: Artifacts must not contain these categories to be automatically assigned to the lifecycle process.
   c. **In Domains**: Select which domains lifecycle process will be used in.

> In cases where two or more automatically assigned lifecycle processes apply for an artifact type, Systinet uses these association rules to assign the most appropriate lifecycle process. See

"Automatic Assignment" in the Concepts Guide for more information about Automatic Assignment.

5. Click **Save**.

How to Define Stages

Lifecycle stages represent important milestones in the lifecycle process. Lifecycle stages and their order are defined in the lifecycle process definition. New lifecycle processes have no stages assigned to them.

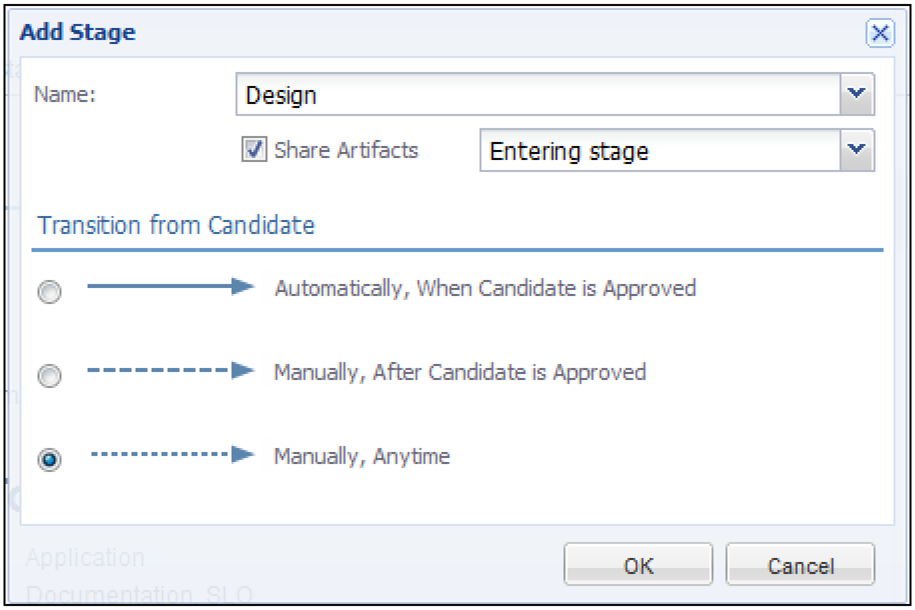
To define lifecycle stages:

1. In the Administration menu of the Administration interface, click **Lifecycle Processes** to open the Lifecycle Processes page, and then click the name of the required process.

   The Lifecycle Processes page opens in the **Overview** tab.

2. To set a initial stage definition for artifacts click **Add Stage**. The Add Initial Stage dialog opens.

Screenshot: Add Stage Dialog



Add Stage

Name: Design

☑ Share Artifacts: Entering stage

Transition from Candidate

○ Automatically, When Candidate is Approved

○ Manually, After Candidate is Approved

● Manually, Anytime

3. Select Stage name from dropdown list or the user may enter a custom name.

   Click the **Share Artifacts** radio button and select when the artifact will be shared if required (this can also be altered in the **Permissions** tab). The *After Approval* option is not available if the transition to the next stage is automatic.

Click **OK**

Additional stages can be added by clicking **Add Stage**, selecting a Stage name and a transition type.

Any process can be edited, cloned, exported, deprecated or deleted by clicking the appropriate button

It is also possible to add the same stage into a process multiple times. These are called referral stages and are visually different from regular stages. Attempting to edit a referral stage will automatically send the user back the properties of the initial version of that stage.

4. Before publishing the process, go to the Stages tab and do the following:

   Note: A valid stage definition must contain at least one of the following conditions.

   - Set transitions

     For details, see "How to Define Transitions" on the next page

   - Set voters for approval

     For details, see "How to Define Approvers" on page 57.

   - Define tasks to complete before stage approval.

     For details, see "How to Define Tasks" on page 56

   - Set policies to comply with before stage approval.

     For details, see "How to Define Policies" on page 56

   - Define automatic actions to execute when a lifecycle transition occurs.

     For details, see "How to Define Automatic Actions" on page 59"

   - Specify whether to automatically promote on stage approval

     For details, see"How to Define Permissions" on page 60

   Click **Save** to save the Stage Definition and return to the Lifecycle Process page.

   All stage details are listed. You can edit or copy a stage definition using **Edit** or **Copy**. Stages can also be deleted by clicking the **Delete** link.

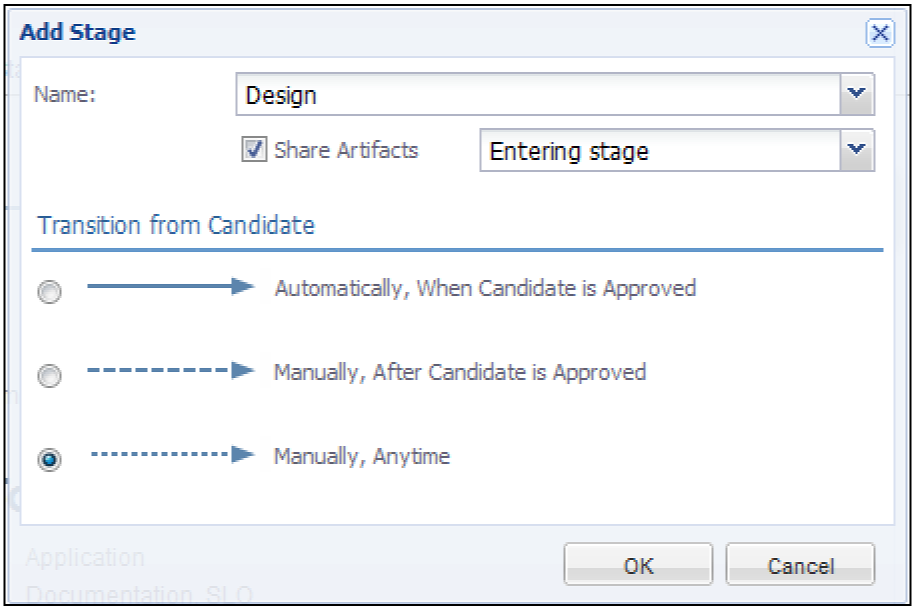
   When process is complete, click **Publish** to make active.

   Note: Stage layout cannot be changed once the process is published.

How to Define Transitions

When adding a stage a type of transition is required by Systinet and there are three options to choose from.

Screenshot: Transition Options



Add Stage

Name: Design

☑ Share Artifacts — Entering stage

Transition from Candidate

○ Automatically, When Candidate is Approved

○ Manually, After Candidate is Approved

● Manually, Anytime

Automatically, When stage_name is Approved

- Stage will transition to the next stage as soon as approved by designated approvers.

Manually, After stage_name is Approved

- Stage will require approval from all designated approvers before it can be manually advanced to the next stage.

Manually, Anytime

- Stage does not require approval from all designated users before being manually advanced

Hovering over the stage or the transition between stages and clicking the pencil icon allows you to edit a transition.

Note: If a stage shares several common previous stages, the transitions types are presented together when editing the stage.

Caution: If a stage has several next stages, only one of the transitions may be automatic and the rest must be set as **Manually, Anytime**. In case of conflict the user will be notified about the automatic change.

How to Define Tasks

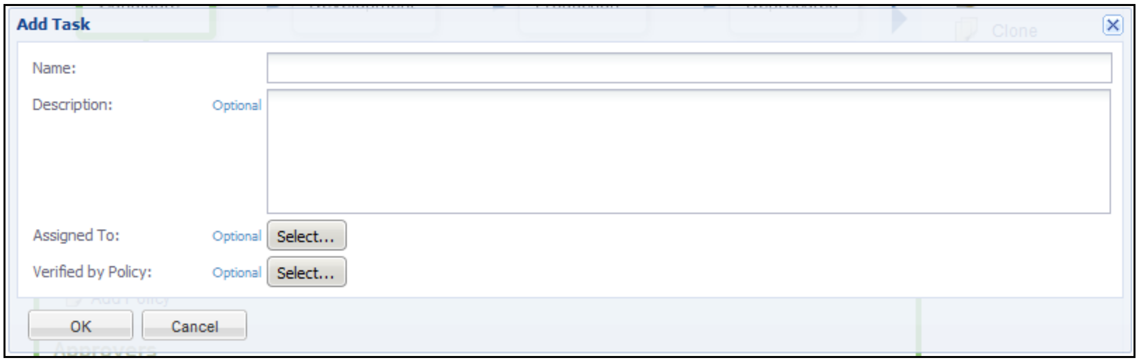
You can define a set of manual tasks to be performed as part of a lifecycle stage.

To add a task:

1. Click the **Stages** tab to enter the **Stage Definition** window and click on the stage that you would like to add a task to.

2. Click **Add Task** to open the Add Task dialog box.

   Screenshot: Add Task Dialog

3. Enter a name and optional description for the task.

4. A Task may be assigned to a Role, a User or a Group by clicking the **Select** next to **Assigned To:** label. Clicking the **Browse Addressbook** button will reveal lists of Roles, Users and Groups that can have assigned tasks.

5. A Task may be verified by policy by clicking the **Select** next to **Verified by Policy** label which will provide a list of existing policies that may be applied.

   Note: Only one policy may be associated with a task at a time. Only policies not used within the same stage can be selected (either in tasks or policies).

6. Click **Ok**.

   The new task is added to the Tasks field.

How to Define Policies

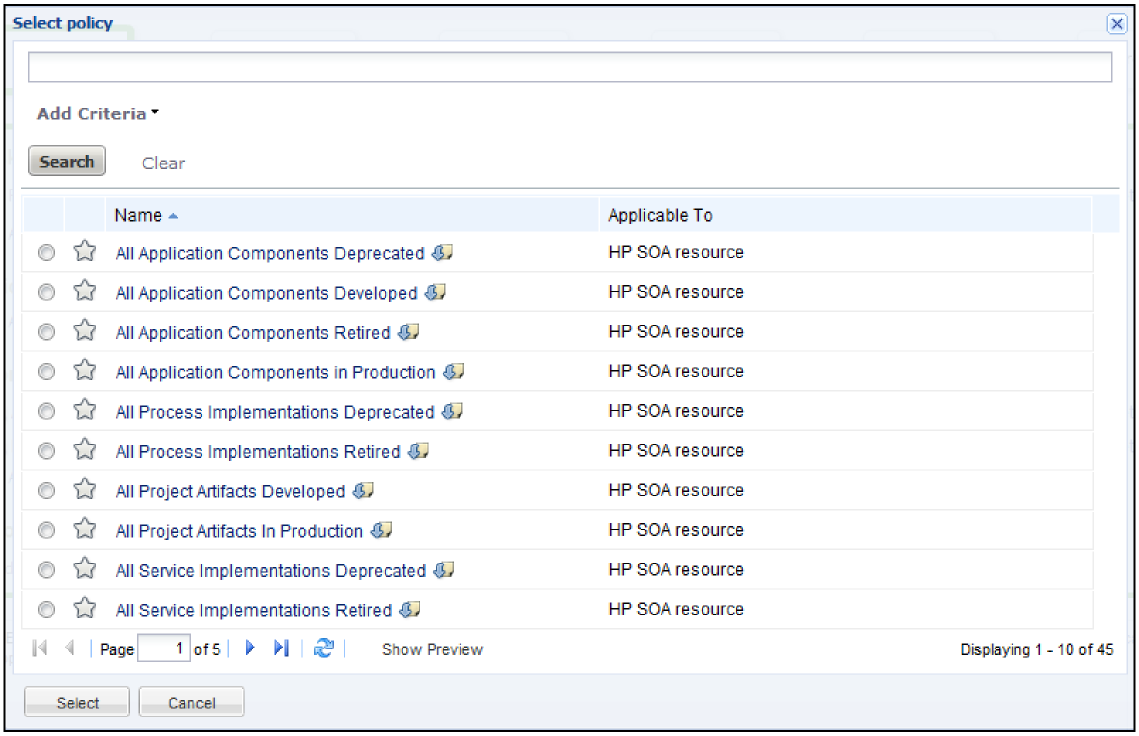
You can define a set of policies to validate artifact compliance as part of a lifecycle stage.

To add a policy:

1. Click **Stages** to open the **Stage Definition** window and click on the stage that requires a policy.

2. Click **Add Policy** to open the **Add Policy** dialog box.

Screenshot: Add Policy Dialog



		Name ▲	Applicable To
○	☆	All Application Components Deprecated	HP SOA resource
○	☆	All Application Components Developed	HP SOA resource
○	☆	All Application Components Retired	HP SOA resource
○	☆	All Application Components in Production	HP SOA resource
○	☆	All Process Implementations Deprecated	HP SOA resource
○	☆	All Process Implementations Retired	HP SOA resource
○	☆	All Project Artifacts Developed	HP SOA resource
○	☆	All Project Artifacts In Production	HP SOA resource
○	☆	All Service Implementations Deprecated	HP SOA resource
○	☆	All Service Implementations Retired	HP SOA resource

You may use either the search function or select a policy from the presented list using the radio button.

Note: Only policies not used within the same stage can be selected (either in tasks or policies).

3. Click **Select**.

   The next windows allows you to change your choice of policy, the artifact type it is applicable to and whether the policy is required to approve the stage.

4. Click **Ok** to apply the policy.

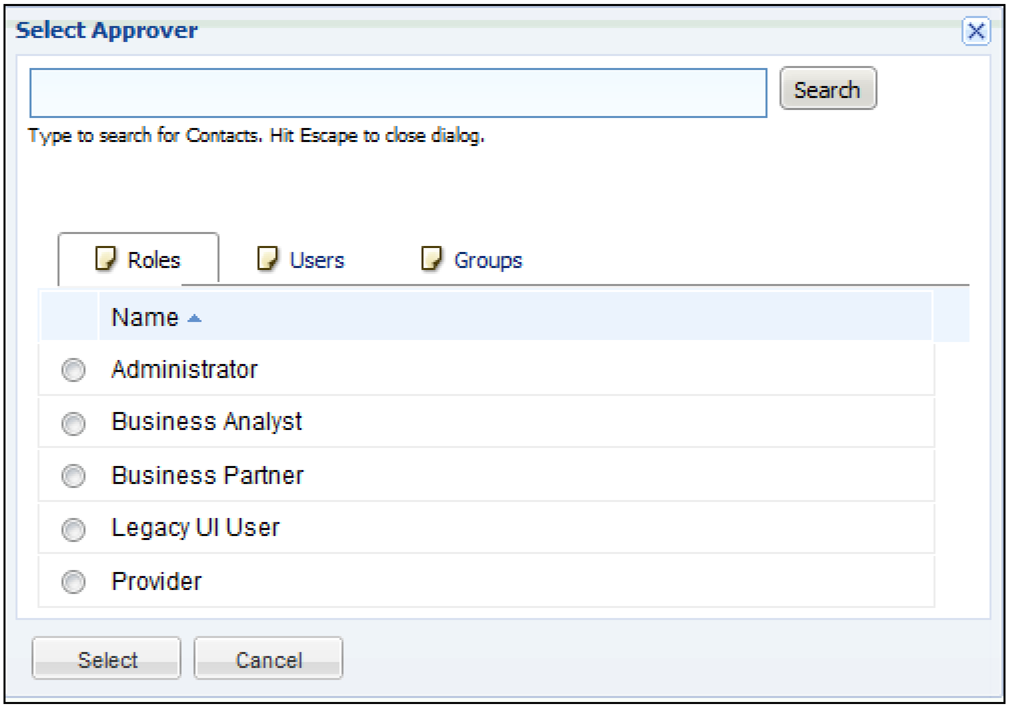
How to Define Approvers

For each lifecycle stage in a lifecycle process you can define a unique set of approvers. The approvers determine whether the artifacts governed by the process are complete for the current stage.

To define voters for approval:

- Click the **Stages** tab on the **Lifecycle Processes** page and then select the stage you require an approval.

Screenshot: Select Approver Dialog



Type to search for Contacts. Hit Escape to close dialog.

	Name ▲
○	Administrator
○	Business Analyst
○	Business Partner
○	Legacy UI User
○	Provider

- To add an individual, group or role of voter(s):

  a. Click the **Browse Addressbook** button.

  b. Principals can be found in the lists or can be found by entering a name in the **Search** field and clicking **Search** to show matching search results.

  c. Click the radio button next to the Role/User/Group required and click **Select** to add selection as an approver.

  d. If the selected approver is not an individual, but a role or group, select the number of votes required by that role or group.

  e. The Passive Approval option can now be selected with a specified number of days before automatic approval.

  f. Click **OK** to add the Approver.

- To remove voters, click on the **Delete** button next to their name

- To edit the Passive Approval option, click the **Edit** button next to the name of the approver.

Note: Approvers can also be viewed, edited or added under the **Permissions** tab.

How to Define Automatic Actions

To define automatic actions, click on the **Add Automatic Actions** link at the bottom of the stage details window on the stages tab.

Dependent on the artifact type, the following actions may be available:

- **Execute Script** - Invoke a custom javascript code previously defined using administration, customization, and manage scripts. In the Advanced tab, you can define a javascript fragment, which is executed prior to running the referenced script artifact.

- **Deploy/Undeploy Contract** - Deploy or undeploy a contract. Deploy action also updates an existing proxy if available. For example, you can deploy a contract to a Layer 7 server when a contract in the Production stage is approved.

- **Update Proxies** - Rebuild PEP proxies for contracts of the approved service or contract provider. For example, the proxies of contracts associated with the governed artifact will be recreated when a contract in the Production stage is approved.

- **Parent Request Approval** - Create approval request for parent artifact. Parent Artifact Type and Parent Stage must be selected in the pop-up dialog. Click **OK** to add automatic action. The approval on parent's artifact will be requested automatically.

- **Enable Consumption** - Allow the creation of new contracts.

- **Disable Consumption** - Prevent the creation of new contracts.

- **Remove Comments** - Once artifact goes into production, all comments will be removed.

- **Delete from Registries** - Will perform delete operation from registry with the possibility of setting when the action is applied. Possible values are Entering stage, Leaving stage and Approval succeeded.

- **Export to Registries** - Will perform export function to registry with the possibility of setting when the action is applied. Possible values are Entering stage, Leaving stage and Approval succeeded.

- **Notify** - Systinet Enables you to send e-mail notifications to artifact stakeholders. You can send set up automatic notifications to be sent as a result of lifecycle changes. Notifications require a subject, and may include Lifecycle Status and an attached text.

The possible recipients of the notification are described in the following table:

Notification Recipients

Recipient	Description
Owner	The user, group, or role that owns the artifact. For details, see "How to Edit Artifact Ownership" in the *User Guide*.
Maintainers	Users, groups, and roles groups with write permission for the artifact. For details, see "How to Edit Access Rights" in the *User Guide*.
Contacts	Users and organizational units associated with the artifact by the contact relationship or listed in the artifact stakeholder property. For details, see "How to Manage Contacts" in the *User Guide*. The Contacts recipient group can be refined into selectable contact roles.
Consumers	People and organizational units with contracts to use the artifact. For details, see "Contracts Overview" in the *User Guide*.
Providers	People and organizational units responsible for contracts associated with the artifacts. For details, see "Contracts Overview" in the *User Guide*.
Other Recipients	Click **Add Other Recipients** and use the user, group, and role search feature to add any other required recipients.
Previous Stakeholders	Expand **Show Advanced Options** and select **Include Recipients from Previous Versions** to notify stakeholders from previous versions of the artifact about changes to a newer version. The stakeholders notified by this option match those for the current artifact version. For example, if **Consumers** is selected, then the consumers of previous versions are notified.

Automatic actions can be deleted by clicking the **Delete** icon and the Notifications and Parent Request Approval actions can be edited by clicking the **Edit** icon.

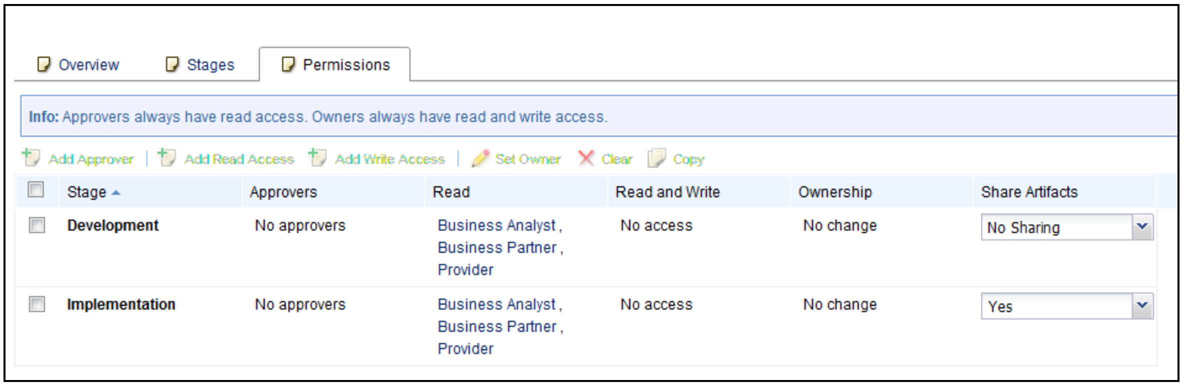
How to Define Permissions

For each lifecycle stage in a lifecycle process you can define a set of permissions. These permissions determine which Role, Group or User has ownership, access rights, approver status and whether artifacts are shared or not.

Note: Permission settings are not carried forward to the next stage.

To set permissions for a stage, select the **Permissions** tab in the **Lifecycle Process** window.

Screenshot: Permissions Matrix



Info: Approvers always have read access. Owners always have read and write access.

□	Stage ▲	Approvers	Read	Read and Write	Ownership	Share Artifacts
□	**Development**	No approvers	Business Analyst , Business Partner , Provider	No access	No change	No Sharing
□	**Implementation**	No approvers	Business Analyst , Business Partner , Provider	No access	No change	Yes

Click the select box for any stage that requires a change in **Permissions** setting.

Add Approver

- Click **Add Approver** to open the Add Approver dialog box

Approvers can be assigned from either the Roles, Users or Groups lists

- Click **Select**

The Approver can be given passive approval with a set date if required

- Click **OK**

Read or Write Access can be assigned to individual Users, Roles or Groups by clicking on the Add Read Access or Add Write Access buttons or the corresponding value in the permissions matrix.

- The owner of the Stage can be assigned by clicking the **Set Owner** button or the **Ownership** value in the matrix.

- Artifacts can be shared by selecting a value from the **Share Artifacts** drop-down list.

- Permissions for a stage can be removed by clicking the **Clear** button

- Defined Stage Permissions can be copied to another stage by clicking the **Copy** button and selecting which stages and/or roles require the copies values.

How to Publish a Process

After creating a lifecycle process, the next step is to make it available for the governance of artifacts.

To publish a lifecycle process:

- From the Lifecycle Processes page, select the processes to be published by clicking on the check box next to the process name and click **Publish**.

- Processes can also be published from the Overview or Stages tab of a selected process.

- Processes can be deprecated by clicking the **Deprecate** button in either the processes page or the Overview or Stages tab of a selected process.

This lifecycle process is now available to be used in the governance of root and sub-artifacts.

How to Export Lifecycle Processes

Any Lifecycle Process can be exported from either the Overview or Stages tab.

- Click **Export** from the right hand menu.

Provide a name for the exported file and click **Export** in the dialog window.

Either open or provide a location for the exported file.

Lifecycle Best Practice

The following list enumerates basic best practice.

- Govern all artifacts
  HP recommends governing artifacts whenever possible. We are aware that in some situations (e.g. for system artifacts) there may be additional overhead, but the benefits gained are worth it. For example, versioning works properly only for governed artifacts.

- Assign lifecycle processes automatically
  We recommend marking lifecycle processes as automatically assignable. In our opinion, common users should not select a lifecycle process for their artifact(s). Instead, the Administrator should create an appropriate lifecycle process that is assigned to the artifact automatically after artifact creation.

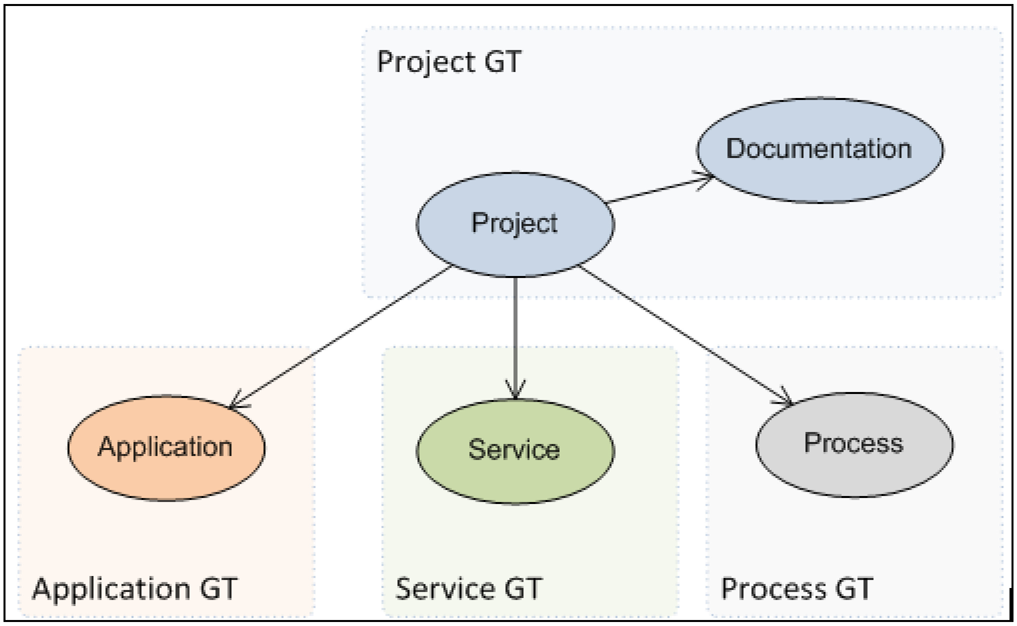
  Action Set Lifecycle Process / Stage provides users with the possibility to set any process to the artifact. Administrators can even set a specific stage and mark it as approved. This action is intended mainly for administrators who import data to the repository. If the data is suitable, Administrators can use this action to set for example, Production stage.

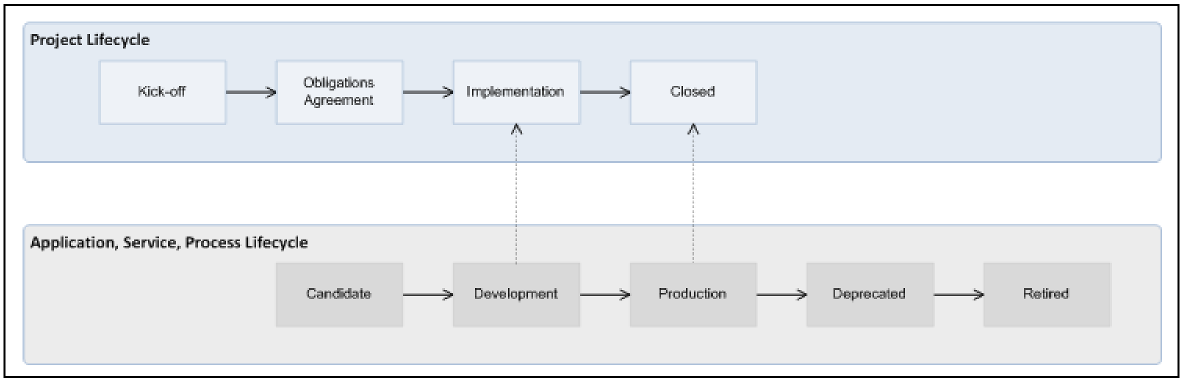
- Separate lifecycle processes
  We recommend separating lifecycle processes and defining different lifecycle process for different artifact types. These processes can be connected together via policies but it is crucial to not govern all artifacts in one process.

As an example, you can review the default lifecycle processes include in this topic. There is a special lifecycle process for Project artifacts containing only Project artifacts and their documents. And similarly there are separated lifecycle processes for Applications, Services, and Processes.

As mentioned, processes are separated but they can be tied together. For example it is possible to define the following rules.

- A Project can be approved at the Implementation stage only if all project artifacts (Applications, Services or Processes) are approved at the Development stage. Similarly, a Project can be approved at the Closed stage only if all underlying artifacts (such as applications, services and processes) are approved at the Production stage.



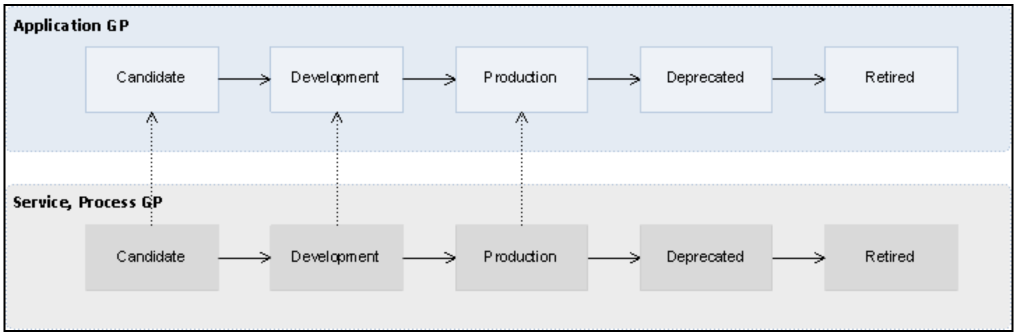


Application Lifecycle

Applications represent business functionality in the real world. Applications can consist of multiple components (services and business processes) but these do not form part of the application lifecycle tree.

Application Lifecycle Tree:

- Application
  - Documentation
  - SLO



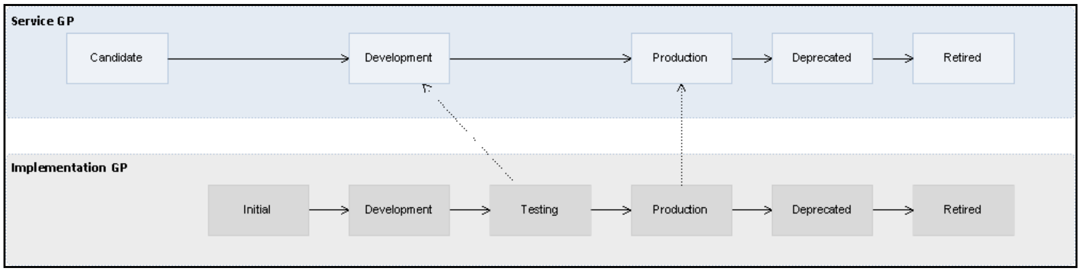
The Application Lifecycle Process depends on the lifecycle processes of its component artifacts. For example, the Production stage of an application cannot be approved until all its component artifacts are approved in the Production stage.

Service Lifecycle

Services represent business functionality (or concepts) in the real world. Services consist of sub-services and implementations but these do not form part of the lifecycle tree.

Service Lifecycle Tree:

- Service
  - Documentation
  - SLO



The Service Lifecycle Process depends on the lifecycle processes of its constituent artifacts. For example, the Development stage of a service cannot be approved until all its sub-services are approved in the Development stage and at least one of its implementations is approved in the Testing stage.

Service Implementation Lifecycle

In the real world an implementation represents a specific service version being implemented, in the repository it represents a package with the service executables which can be later deployed to multiple environments.

Service Lifecycle Tree:

- Implementation
  - Operation
  - Endpoint
  - Interface (WSDL)
  - Schema (XSD)
  - Documentation
  - SLO

Each implementation represents one service and the Service Lifecycle Process depends on the Implementation Lifecycle Process. For details, see "Service Lifecycle" on the previous page

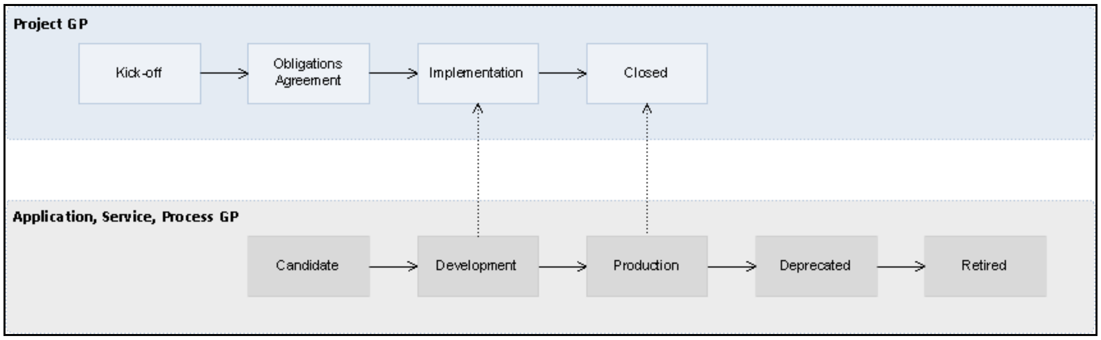
Project Lifecycle

Projects represent any progressive activity (e.g. service development). Projects can be composed of services, applications, business processes, and other artifacts. In this release, the goal of a project is to successfully deploy its constituent artifacts to a production environment.

Project Lifecycle Tree:

- Project
  - Documentation

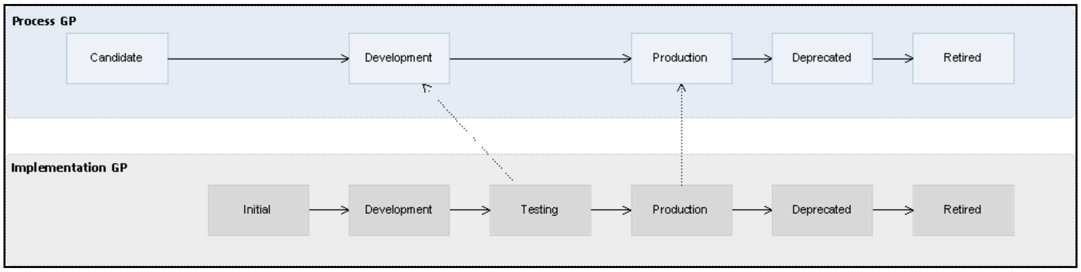
Promotion is automatic after the approval of each stage as per the following image:



The Project Lifecycle Process depends on the lifecycle processes of its constituent artifacts (applications, services, and business processes). For example, the Closed stage of a project cannot be approved until all its constituent artifacts are approved in the Production stage.

Process Lifecycle

The Process lifecycle is similar to the Service Lifecycle. Some of the tasks and policies vary, but the stages and dependency on the process implementation lifecycle is the same as that between a service and its implementation.



Process Implementation Lifecycle

The lifecycle process for business process implementations is similar to that for service implementations. Some tasks and policies vary, but the stages and relationships to the process lifecycle are the same.

Chapter 8: Product Integration Management

To enable product integration in Systinet, the administrator must create server artifacts representing the integrated servers and perform some configuration tasks. For more details about product integration, see "Product Integration Overview" in the *User Guide*.

Access server management from the Administration tab->Administration menu. Click **Servers** to open the Servers tab in the current domain.

> **Note:** UDDI and ALM servers are associated with the domain which you are using when you create them. There can only be one instance of BSM / UCMDB and they are associated with the top-level domain, regardless of the domain you are using when you create the server.

The administrator performs the following tasks:

- "How to Manage Integrated Products" below
- "How to Synchronize Registry Taxonomies" on the next page

"Integration UI Reference" on page 72 provides a guide to the input parameters for creating integrated server artifacts.

How to Manage Integrated Products

Systinet enables you to integrate with other HP products.

To Add a Server for an Integrated Product:

1. To access server management, in the Administration tab-> Administration menu, click **Integrations** to open the Domain details page for the current domain in the Integrations tab.

2. Click **Add Server** under RGIF Devices for the product you want to integrate. The Create *Product* Server page opens for the selected product. For parameter details, see the following topics:

   - "Create UDDI Registry Server Page" on page 74
   - "Create BSM / UCMDB Server Page" on page 72
   - "Create ALM 11.5x Server Page"

   > **Note:** UDDI and ALM servers are domain specific, create these server artifacts in the working domains where users need to access them. You can only integrate one BSM server and you must create these server artifacts in the top-level domain.

For details about the functionality available for each product integration, see the following topics in the *User Guide*:

- "UDDI Registry Integration"
- "BSM / UCMDB Integration"
- "ALM Integration"

How to Synchronize Registry Taxonomies

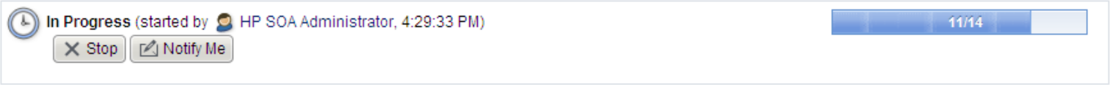
A general precondition must be fulfilled before registry import/export. Taxonomies with the taxonomic values referenced by the imported/exported data must be present in both the source and target environment. The easiest way is to synchronize the taxonomies between the registry and Systinet. In the case of missing taxonomies during import/export,Systinet outputs an error message about the missing taxonomies and prevents the import/export. If the registry is an HP Registry, you can perform taxonomy synchronization directly from the Administration tab.

Note: Only a user with the administrator role can perform taxonomy synchronization. Systinet controls the availability of Import and Export Taxonomy functionality based on the Allow Import / Export settings used when the server artifact is created.

To Import Taxonomies from a Registry:

1. In the Artifact Detail page of the registry, click the **Import Taxonomy** context action to open the **Import Taxonomy** dialog box.
2. *Optional:* Click **(Change)** to use different Registry credentials.
3. Select to show **Missing Taxonomies** or **All**.
4. Use the filter to locate specific taxonomies to import, select the taxonomies to import, and then click **Import**.

This process executes as a bulk operation. An information bar opens informing you that the operation is in progress with a progress bar with options to **Stop** the operation or to **Notify Me** when the operation is complete.

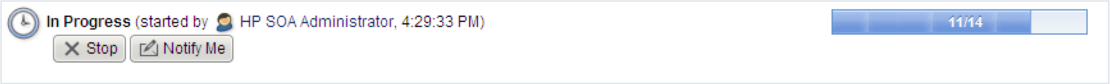

In Progress (started by HP SOA Administrator, 4:29:33 PM)

The operation executes asynchronously, so you can navigate away and perform other tasks while the operation completes. For more details, see "How to Use Bulk Operations" in the *User Guide*.

To Export Taxonomies to a Registry:

1. In the Artifact Detail page of the registry, click the **Export Taxonomy** context action to open the **Export Taxonomy** dialog box.
2. *Optional:* Click **(Change)** to use different Registry credentials.
3. Select to show **Missing Taxonomies** or **All**.
4. Use the filter to locate specific taxonomies to export, select the taxonomies to export, and then click **Export**.

This process executes as a bulk operation. An information bar opens informing you that the operation is in progress with a progress bar with options to **Stop** the operation or to **Notify Me** when the operation is complete.


In Progress (started by HP SOA Administrator, 4:29:33 PM)

The operation executes asynchronously, so you can navigate away and perform other tasks while the operation completes. For more details, see "How to Use Bulk Operations" in the *User Guide*.

How to Configure BSM/UCMDB Integration Mapping

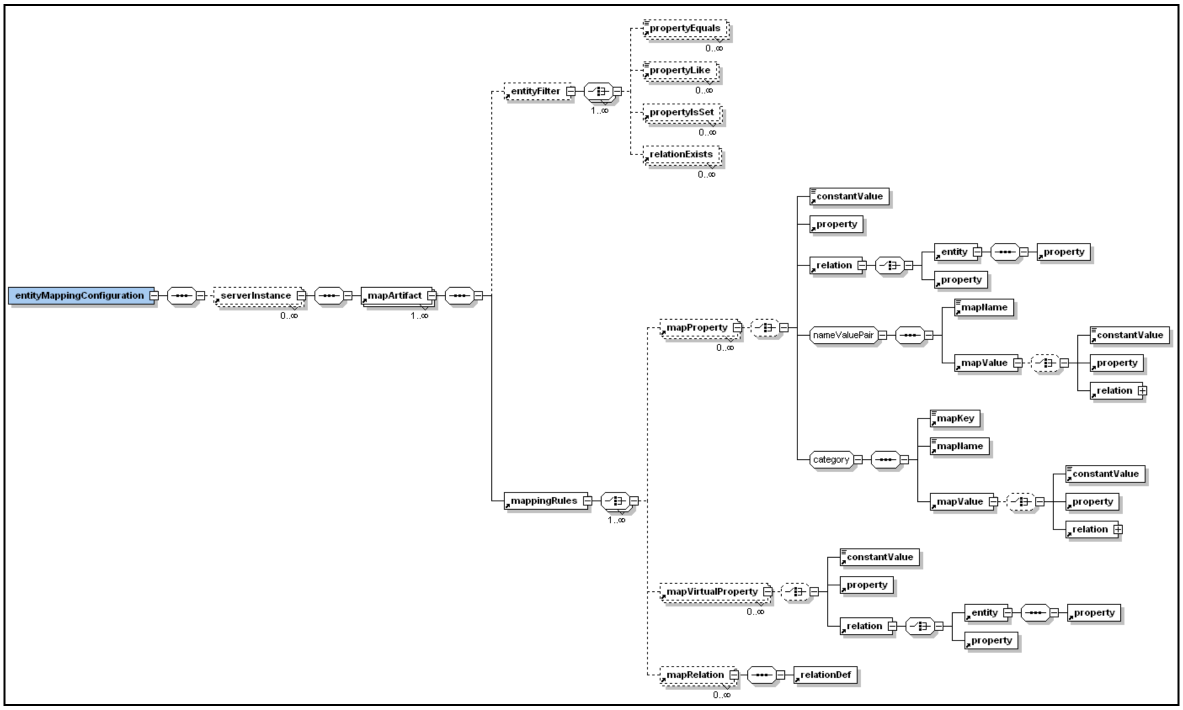
Introduction

Configuration of integration between HP Systinet and BSM/UCMDB is used to:

- Know which Configuration Items (CI) in CMDB (Configuration Management Database) to import.
- Specify item type with optional entityFilters that can further restrict imported data by property equals/like/isset and relation exists.
- Know with attributes of a particular configuration item to load.
- Provide mapping to Systinet artifacts and its properties.

This configuration is stored in XML format and conforms to **entity-mapping-1.0.xsd** (you can find this in [SYSTINET_HOME]/lib/in-discovery.jar/META-INF/entity-mapping-1.0.xsd)

Following is a visual view of this XSD:



This configuration is stored in **System Settings** of HP Systinet with name **platform.integration.discovery.bac.mappingConfiguration***

Default configuration

In default configuration provided by HP Systinet, there are three Configuration Item templates which are mapped to Systinet Definition Model (SDM).

Configuration Item template	SDM Artifact
webservice	SOAP Service
business_service_for_catalog	Business Service
business_unit	Organization Unit

Artifact: webService

Mapped to	Mapped from
artifact **urn:com:systinet:soa:model:artifacts:soa:implementation:ws**	entity **webservice**
property **urn:com:systinet:soa:model:properties:name**	property **display_label**
property **urn:com:systinet:soa:model:properties:description**	property **data_description**
property **urn:com:systinet:soa:model:properties:serviceName**	property **service_name**

property **urn:com:systinet:soa:model:properties:serviceNamespace**	property **data_name**
property **urn:com:systinet:soa:model:properties:definition**	additionally filled during WSDL import
relation **urn:com:systinet:soa:model:properties:wsdlPortRelationship**	additionally filled during WSDL import
relation **urn:com:systinet:soa:model:properties:endpoint**	additionally filled during WSDL import
virtual **property discovery.service.wsdl_url**	property **wsdl_url**
virtual **property discovery.service.host**	property **data_name** of entity host through relation **depends_on**

Artifact: businessService

Mapped to	**Mapped from**
artifact **urn:com:systinet:soa:model:artifacts:soa:businessService**	entity **business_service_for_catalog**
property **urn:com:systinet:soa:model:properties:name**	property **display_name**
property **urn:com:systinet:soa:model:properties:description**	property **data_description**
relation **urn:com:systinet:soa:model:properties:service**	relation **depends_on** to webservice

Artifact: organizationUnit

Mapped to	**Mapped from**
artifact **urn:com:systinet:soa:model:artifacts:content:contact:organizationUnit**	entity **business_unit**
property **urn:com:systinet:soa:model:properties:name**	property **display_name**
property **urn:com:systinet:soa:model:properties:description**	property **data_description**
relation **urn:com:systinet:soa:model:properties:provides**	relation **contains** to webservice
relation **urn:com:systinet:soa:model:properties:provides**	relation **provider_of** to business_service_for_catalog

> **Note:** Virtual property is persisted to **integrationProperty** of **ExternalEntityArtifact**. These properties are necessary for extra processing such as file publishing and metrics inquiry.

Integration UI Reference

This chapter provides a reference to server creation and edit pages that are accessible from the Domains page Servers tab in the Administration UI:

- "Create BSM / UCMDB Server Page" below
- "Create ALM 11.5x Server Page" on the next page
- "Create UDDI Registry Server Page" on page 74

Create BSM / UCMDB Server Page

The Create and Edit BSM / UCMDB Server pages contain the following parameters:

Parameter	Definition
Name	Name of the BSM / UCMDB server artifact.
Description	A description including rich text and HTML support. This description displays in the Artifact Details page Overview tab properties area. For details, see *User Guide* "Artifact Details Page Overview Tab" in the . This description is limited by the display length available in the Overview tab. If you need a longer description or an artifact specification edit the Specification property. For details, see "How to Edit Artifact Specifications" in the *User Guide* .
Base URL	The address of the BSM/UCMDB server. For example `http://mybsmserver.com/topaz`.
Username and Password	Credentials to access the BSM server, required to access service availability information from the BSM server and with an option to save these credentials for default use.
UCMDB Username and Password	Credentials to access the UCMDB server, required for service discovery.
Environment	Select an environment to associate with the server. For details, see "Environments" in the *User Guide*. The environment setting categorizes endpoints during service discovery.
Publishing Location	Repository location to store resource content.

Important: In order for the HP Systinet server to connect with the BSM/UCMDB server via HTTPS protocol, you will need to import the certificate of BSM into HP Systinet.

How to import the BSM certificate into HP Systinet:

1. Access the URL of the BSM/UCMDB server via web browser (HTPPS protocol). The web browser will ask you to import a certificate.

2. Export the certificate from the web browser (for example, export the certificate into bsm.cert).

3. Run the following command:

   ```
   keytool -import -alias myBSMServer -file bsm.cert -keystore SYSTINET_
   HOME/conf/client.truststore
   ```

Note: User must provide a trust store password is **changeit** during step 3.

Create ALM 11.5x Server Page

The Create and Edit ALM Server pages contain the following parameters:

Parameter	Definition
Name	Name of the ALM server artifact.
Description	A description including rich text and HTML support. This description displays in the Artifact Details page Overview tab properties area. For details, see *User Guide* "Artifact Details Page Overview Tab" in the . This description is limited by the display length available in the Overview tab. If you need a longer description or an artifact specification edit the Specification property. For details, see "How to Edit Artifact Specifications" in the *User Guide* .
Base URL	The address of the ALM server. For example, ALM 11.5x:`http:// myalmserver.com:8080/qcbin/rest`.
Username and Password	Credentials to access the ALM server with an option to save them for default use.
Server Version	ALM 11.5x. The Base URL field format.
ALM Domain	The domain containing the testing project in ALM.
ALM Project	The testing project in ALM.

Parameter	Definition
Environment	Select an environment to associate with the server. For details, see "Environments" in the *User Guide*. The environment setting is used for export to ALM functionality. If the ALM server is categorized, when a service is exported, the associated WSDL added to ALM is amended to contain any related matching environment categorized endpoints.

Create ALM 12 Server Page

The Create and Edit ALM Server pages contain the following parameters:

Parameter	Definition
Name	Name of the ALM server artifact.
Description	A description including rich text and HTML support. This description displays in the Artifact Details page Overview tab properties area. For details, see *User Guide* "Artifact Details Page Overview Tab" in the . This description is limited by the display length available in the Overview tab. If you need a longer description or an artifact specification edit the Specification property. For details, see "How to Edit Artifact Specifications" in the *User Guide* .
Base URL	The address of the ALM server. For example, ALM 12.00:`http:// myalmserver.com:8080/qcbin/rest`.
Username and Password	Credentials to access the ALM server with an option to save them for default use.
Environment	Select an environment to associate with the server. For details, see "Environments" in the *User Guide*. The environment setting is used for export to ALM functionality. If the ALM server is categorized, when a service is exported, the associated WSDL added to ALM is amended to contain any related matching environment categorized endpoints.

Create UDDI Registry Server Page

The Create and Edit UDDI Registry Server pages are split into two pages containing the following parameters:

Parameter	Definition
Hostname	The name of the server where the registry is running.
HTTP Port	The port for non-secure access to the registry.

Parameter	Definition
SSL (HTTPS) Port	The port for secure access to the registry.
Application Server Context	If the registry is ported to a J2EE server then input the context name.
HP SOA Registry	Select if the registry is HP SOA Registry Foundation.

Parameter	Definition
Name	Name of the registry artifact.
Description	A description including rich text and HTML support. This description displays in the Artifact Details page Overview tab properties area. For details, see *User Guide* "Artifact Details Page Overview Tab" in the . This description is limited by the display length available in the Overview tab. If you need a longer description or an artifact specification edit the Specification property. For details, see "How to Edit Artifact Specifications" in the *User Guide* .
Username and Password	Credentials to access the registry. These credentials are stored against the server as the default credentials.
Credential Usage	Select the use of the default credentials. If **Import Only** is selected then users must provide their own credentials to be able to export and delete.
HP SOA Registry	Select if the registry is HP SOA Registry Foundation.
Allow Import	Select if the registry is a source for data import.
Allow Export	Select if the registry is a target for data export.
HTTP Basic Authentication	Select if the registry is secured with HTTP Basic Authentication.
Environment	Select an environment to associate with the server. For details, see "Environments" in the *User Guide*. The environment settings determine registry import and export functionality for endpoints.
Publishing Location	Repository location to store resource content.
Server URLs	The set of connection URLs for the registry.

Chapter 9: Configuration Management

Systinet provides a customizable configuration that enables the administrator to control nearly every aspect of the behavior of Systinet.

The Administrator can modify most parts of the configuration of Systinet from the Administration tab.

In the Administration menu, select **Configuration** to open the Configuration page.

The Configuration page is split into the following tabs:

- **Basic Settings**

  Settings that affect the operation of Systinet repository. For more details, see "How to Manage Basic Configuration Options" on the next page.

- **System Settings**

  The detail level settings of the configuration. The administrator can use this tab to modify individual settings at a global, or domain level. This tab also enables the export and import of the configuration as a whole. For details, see "How to Manage System Settings" on page 81.

- **License**

  A summary page detailing the terms of the current license with an option to enter a new license key and manage licensed users. For more details, see "License Management" on page 78.

- **Self-Test**

  Details the status of Systinet with an option to disable self-test. For more details, see "Self-Test" on page 80.

- **Artifact Validation**

  Configure policies to validate artifacts when users create or edit them. For details, see "How to Manage Artifact Form Validation" on page 102.

- **Report Cleanup**

  Set the maximum age of reports and events to be retained when the Report Cleaner Task executes. You can set a different age by report and event type. You can also use the following system properties to filter the processing of the Report Cleaner Task:

  - `platform.reportCleanerDao.batchNum` - Sets the number of reports to be cleaned in a database transaction. The default value is 20.

  - `platform.reportCleanerDao.interval` - Sets the interval (in millisecond) that the Report Cleaner Task will sleep after a database transaction is committed. The default value is 10000.

Manage the execution and scheduling of the Report Cleaner Task using the Tasks page. For details, see "Administration Task Management" on page 104.

Other aspects of the configuration are determined during installation and the administrator can change them using the Setup Tool. For details, see "Setup Tool" on page 249.

How to Manage Basic Configuration Options

Systinet enables the administrator to control some basic aspects of the configuration from the Administration tab.

To Configure Basic Settings:

1. In the Administration menu, select **Configuration** to open the Configuration page in the Basic Settings tab.

2. In the Basic Settings tab, set any of the following options:

   - **Full Text Search**

     Select this option to enable full text search in the Systinet UI.

     > **Note:** Full-text search must also be enabled on the database as described in the following sections of the *Installation and Deployment Guide*:
     >
     > - "Enable Full-Text Search in DB2"
     > - "Enable Full-Text Search in MSSQL"
     > - "Enable Full-Text Search in Oracle"

     By default, Systinet appends a % to search terms. To disable this functionality after installation, set the configuration property `shared.db.fulltextsearch.appendpercentage` to `FALSE`. For details, see "How to Manage System Settings" on page 81.

   - **SSL Customization**

     Select the authentication method to apply to SSL certificates.

     Certificate Trust and Customization Categories

Certificate Trust	Customization
Java/JSSE default key/trust stores...	default
Server certificates are always trusted...	skipped

Certificate Trust and Customization Categories, continued

Certificate Trust	Customization
Database key/trust stores...	database
Composition of database and default...	composite

The options correspond to those described in "SSL Customization" on page 225.

3. Click **Save** to make your changes or **Reset to Defaults** to restore these settings to their defaults.

License Management

Systinet employs a licensing system to control the following aspects of an installation:

- Usage limits, including the following:
  - Time limited licenses
  - Concurrent user limited licenses
  - Licensed user limited licenses
  - Connectors for RGIF device
- Feature limits, including the following:
  - Lifecycle Management
  - Policy Management

Access the license information from the Administration tab. In the Administration menu, click Configuration to open the Configuration page, and select the License tab to view the current license.

The License page provides the functionality described in the following topics:

- "How to Change the License" below
- "How to Manage Licensed Users" on the next page
- "How to Reset the License" on the next page

How to Change the License

The administrator can change the Systinet license.

To Change the License:

1. Contact HP Support by submitting a new service request (for a new license key) at: http://support.openview.hp.com/casemanager/newincident

   Alternatively, you can ask your sales representative for a new license key.

2. In the Administration tab->Administration menu, click **Configuration** to open the Configuration page.

3. Select the **License** tab to view the current license details.

4. Click **Enter License Key** to open the Enter License Key page.

5. Select a license key file, then click **Finish**. If the license contains a user and a license key, then click on **Enter License Key** button and then copy paste them to their respective fields.

   The Configuration page opens in the License tab showing the details of the new license.

How to Manage Licensed Users

Where the license determines a limited number of user seats, the administrator can modify which users can access Systinet.

To Manage Licensed Users:

1. In the Administration tab->Administration menu, click **Configuration** to open the Configuration page.

2. Select the **License** tab to view the current license details.

3. In the License Conditions section, click **Edit** next to the Licensed User Seats to open the Edit Licensed Users page.

   Note: The Edit option is only visible for licenses with limited user seats.

4. In the Edit Licensed Users page, do any of the following:

   - To add licensed users, click **Add** and use Browse Address Book or the search input to locate the users you want to add. Click **Select** to confirm your additions.

   - To remove licensed users, select the users in the table, click **Remove** and confirm your decision.

How to Reset the License

There are two ways in which you can reset the license without having to re-install the product:

To reset the license through the UI:

1. Select the **Administration** tab > **Configuration** > **System Settings** to open the Configuration page.

2. In the Name text box, search for the file named **shared.license.ds1**.

3. Click **Download**, then **Open** to edit the setting on your computer.

4. In System Settings, click **Upload** to upload the edited file.

5. Restart the server for the changes to take effect.

To reset the license directly in the Database (DB):

1. Stop the server.

2. Use the SQL Developer (if you use Oracle) and connect to the **Systinet** database.

3. Open the systemconfiguration table, filter with the name **shared.license.ds1** and edit the **BVALUE** column.

4. In the editor, save the contents as a local file. Edit the local file as desired and then load it to the **BVALUE** column.

5. Commit the change.

6. Restart the server.

Self-Test

Systinet includes a Self-Test application that runs a series of checks on the installation and configuration.

The administrator can access the Self-Test from the Administration tab. In the Administration menu, select **Configuration** to open the Configuration page, and then select the **Self-Test** tab .

The Self-Test tab displays a summary and then the details of the self-test checks.

The self-tester performs the following checks:

Self-Tests

Self-Test	Description
Product configuration checks	Checks product configuration, versions, and libraries.
Product runtime checks	Checks logging configuration and outputs based on product URLs.
Application server checks	Checks application server and JVM settings.
JNDI checks	Checks required JNDI resources.

Self-Tests, continued

Self-Test	Description
Datasource checks	Checks the data source connection.
JMS checks	Checks the sending of JMS messages to required JMS destinations.
LDAP checks	Checks LDAP connectivity, if configured during installation or setup.
Performance	Basic Systinet performance checks.

In the default configuration, the output includes only information about the groups of checks that are run and any errors that occur. The full self-test output is stored in the application server log folder, `systinet_self_test.log`.

Use the **Disable** / **Enable** button to turn Self-Test off and on.

A machine-readable output of the self-test is available at http://hostname:port/context/self-test/xml.

Self-Test is also available as a stand-alone deployment which you can use during installation and deployment. For details, see "Systinet Self-Test" in the *Installation and Deployment Guide*.

How to Manage the System Configuration

Systinet stores system settings in a configuration file in the installation folder. The Administrator can view and edit these settings directly in the Administration tab.

In the Administration menu, click **Configuration** to open the Configuration page, and select the **System Settings** tab to view the current system configuration.

The Configuration page System Settings tab provides functionality described in the following topics:

- "How to Manage System Settings" below
- "How to Export and Import System Settings" on the next page
- "System Configuration Properties" on page 83 provides a reference to configuration properties available in the System Settings tab.

How to Manage System Settings

In the Configuration page System Settings tab, the administrator can add, edit, and remove system settings.

System settings can apply globally, or within a specific domain. If a domain setting exists, it takes precedence over a global setting.

To Add System Settings:

1. In the Configuration page System Settings tab, click **Add Property** to open the Add Property dialog box.

2. *Optional:* Click **Change** to alter the domain that the setting applies to.

3. Set a name and value for the property, and click **OK** to add the property to the system settings for the selected domain.

To Add System Settings from a File:

1. In the System Settings tab, click **Add File Property** to open the Add File Property dialog box.

2. *Optional:* Click **Change** to alter the domain that the setting applies to.

3. Set a name for the property, browse for the file on your local filesystem, and click **OK** to add the property to the system settings for the selected domain.

To Edit System Settings:

1. In the System Settings tab, use the Name filter to locate the setting you want to edit.

2. Click **Edit** for the setting to open the Edit Property dialog box.

3. Do one of the following:

   - Input a new value and click **Save** to set a new value.

   - Click **Load Default Value** to reset the value to its default installation value.

To Delete System Settings:

1. In the System Settings tab, select the properties you want to delete.

2. Click **Remove** and confirm your decision.

How to Export and Import System Settings

The administrator can export system settings to a ZIP file to preserve a particular configuration. Import enables a stored configuration to overwrite the current settings.

To Export System Settings:

1. In the Configuration page System Settings tab, select the properties to export, click **Export**, and confirm your decision.

2. Click **Save** and select a file location for the ZIP file.

To Import System Settings:

1. In the System Settings tab, click **Import** to open the Import dialog box.

2. Input or Browse for the file, and click **OK**.

System Configuration Properties

This reference topic divides the configuration properties by module and specified their type into the following scenarios:

- **SC-I**: Runtime configurable properties that apply immediately.

- **SC-II**: Runtime configurable properties with additional steps required. After the change, the administrator should make some additional changes outside the application, for example, change the application server configuration.

- **SC-III**: Runtime configurable properties which require a restart of the application server.

- **SC-IV**: These properties can only be changed using the Setup Tool.

Usage Statistics

Property Name	Description	Value Type	Default	Scenario
`platform.usage.stats.excluded.artifact.types`	List of artifact types which are excluded from Activity Report (usage stats) computation.	Comma-separated list of artifact types.	*reportArtifact,taskArtifact*	`SC-I`

Security

Property Name	Description	Value Type	Default	Scenario
`shared.siteminder.enabled`	Turns on Siteminder (HTTP Header) authentication.	Boolean value	*false*	`SC-IV`
`shared.siteminder.useCookie`	Sets whether use cookie to pass the login name.	Boolean value	*false*	`SC-I`
`shared.siteminder.useHeader`	Sets whether use HTTP Header to pass the login name.	Boolean value	*true*	`SC-I`
`shared.siteminder.loginNameField`	Sets name of login name header/cookie.	Text value	*sm-userdn*	`SC-I`

Security, continued

Property Name	Description	Value Type	Default	Scenario
`shared.siteminder.importGroups`	Sets whether import also groups to security context.	Boolean value	*false*.	`SC-I`
`shared.siteminder.groupsHeaderName`	Name of groups header or cookies used in case useCookies equal to true.	Text value	*sm-role*	`SC-I`
`shared.siteminder.groupsHeaderDelimiter`	Delimeter for found groups.	Text value	^	`SC-I`
`shared.siteminder.requireAuthentication`	Require authentication by siteminder, it fails when no user is received from siteminder.	Boolean value	*false*.	`SC-I`
`shared.dql.security.allowNoAcl`	Determines whether ACL settings are used for DQL queries.	TRUE or FALSE	FALSE	`SC-III`
`shared.dql.security.allowNative`	Determines whether NATIVE clauses are allowed in DQL queries.	TRUE or FALSE	FALSE	`SC-III`

Versioning

Property Name	Description	Value Type	Default	Scenario
`platform.versioning.schema.initial`	Initial schema version.	Text value	*1.0*	`SC-I`
`platform.ui.newVersion.create.` `blankArtifact`	New version 'Create Blank Artifact' default choice.	Boolean value	*false*.	`SC-I`

HTTP Client

Property Name	Description	Value Type	Default	Scenario
`shared.http.connections.per.host`	Max count of connections per host.	Integer value.	*30*	`SC-I`
`shared.http.total.connections`	Max count of available connections by HTTPClient.	Integer value.	*100*	`SC-I`
`shared.http.user.agent.identification`	Identification of user agent header. Sets user agent header field for all calls which do not specify another one.	Text value.	*HTTPClient/?.?*	`SC-I`
`shared.http.client.proxy.enabled`	Sets whether proxy is enabled.	Boolean value.	*false*.	`SC-I`
`shared.http.client.proxy.nonproxyhosts`	Sets all non proxy hosts.	Text value.	Not used. (*localhost,127.0.0.1*)	`SC-I`
`shared.http.client.proxy.host`	Proxy host.	Text value.	Not used. (*proxyhost*)	`SC-I`
`shared.http.client.proxy.port`	Proxy port.	Text value.	*8080* - Not used.	`SC-I`
`shared.http.client.proxy.user`	If is proxy secured, user name of proxy user.	Text value.	Not used.	`SC-I`
`shared.http.client.proxy.password`	If is proxy secured, password of proxy user.	Text value.	Not used.	`SC-I`

Publishing

Property Name	Description	Value Type	Default	Scenario
`platform.publishing.limit.concurrent.uploads.user`	Maximum count of concurrent uploads by a single user.	Integer value.	*3*	`SC-I`
`platform.publishing.limit.concurrent.uploads.total`	Maximum count of concurrent uploads by all users.	Integer value.	*6*	`SC-I`
`platform.publishing.limit.max.archive.files`	Maximum number of files in an archive file for publishing.	Integer value.	*10000*	`SC-I`
`platform.publishing.limit.max.archive.bytes`	Maximum size of an archive file for publishing.	Integer value.	*10000000*	`SC-I`
`platform.publishing.memory.limit.single`	Max possible amount memory used by single consumer.	Integer value.	*65536*	`SC-I`
`platform.publishing.memory.limit.all`	Max possible amount of usable memory.	Integer value.	*2097152*	`SC-I`
`shared.http.urlbase`	URL base of repository.	URL.	Inserted in installation process.	`SC-IV`

Publishing, continued

Property Name	Description	Value Type	Default	Scenario
`platform.publishing.duplicates.limit`	Max count of duplicities returned by duplicate finder in publisher.	Integer value.	*5*	`SC-I`
`platform.publishing.http.connection.manager.timeout`	HTTP connection manager timeout.	Integer value.	*3000*	`SC-I`
`platform.publishing.http.socket.timeout`	Socket timeout.	Integer value.	*3000*	`SC-I`
`shared.http.urlbase`	Unsecured URL base of repository.	URL.	Inserted in installation process.	`SC-IV`
`shared.https.urlbase`	Secured (SSL) URL base of repository.	URL.	Inserted in installation process.	`SC-IV`
`platform.publishing.ui.zipArchiveExtensions`	A list of archive types that Systinet extracts during publishing.	Comma-separated list of archive extension types that must conform to the ZIP format.	*zip,jar,ear,war,bpr*	`SC-I`

Publishing, continued

Property Name	Description	Value Type	Default	Scenario
`platform.publishing.ui.enforcedCollections`	By default, if publishing fails for a data resource, Systinet publishes it as documentation. Use this property to prevent this default functionality for specified artifact types.	List of collections to enforce during publishing.	*documentation*	`SC-I`
`platform.ui.upload.allowed.extensions.enable`	Set whether to enable upload prevention for malicious files.	Boolean value	False	`SC-I`
`platform.ui.upload.allowed.extensions`	List of extensions allowed for upload.	Comma-separated list of allowed extensions	txt,html,htm,xml,csv,doc, docx,odt,ppt,pps,pptx,odp, xslx,xsl,ods,odg,odf,odb, pdf,svg,jpg,jpeg,png,gif,ico, xslt,dtd,xsd,wsdl,xpdl,bpel, composite,sca,zip,jar,rar,tar,gzip,gz.	`SC-I`

Publishing, continued

Property Name	Description	Value Type	Default	Scenario
`platform.publishing.limit.max.file.bytes`	Maximum size of a file for import (such as wdsl, docx, xslx.) Limit to 2000000 bytes.	Integer value.	2000000 bytes	`SC-I`
`platform.publishing.limit.max.archive.bytes`	Maximum size of an archive file for publishing. Limit to 10000000 bytes.	Integer value.	10000000 bytes	`SC-I`
`platform.publishing.transaction.timeout`	Sets the publishing transaction timeout time.	Integer value.	*-1*	`SC-I`
`platform.publishing.disallowDoctypeDecl`	Does not allow to parse XML file with DTD code inline.	Boolean value	True	`SC-I`

WSDL Publishing

Property Name	Description	Value Type	Default	Scenario
`platform.publishing.ui.defaultDecomposition`	Default decomposition setting on Upload Data Content. 0 - None (Only WSDL), 1 - Implementations (WSDL and related, SOAP S.+Enpoint+Opreations), 2 - As 1 + Business S.	Integer value.	*1*	`SC-I`
`platform.publishing.ui.defaultServiceType`	Default service type setting. One of `[businessService, applicationService, infrastructureService]`	One of service type.	*businessService*	`SC-I`
`platform.publishing.ui.force.duplicate.resolution`	Default setting to publish a duplicate resource to another location.	duplicate, reuse	prompt	`SC-I`

Contract Management

Property Name	Description	Value Type	Default	Scenario
`platform.cm.consumers`	Artifact types defined as contract consumers.	Comma-separated list of artifact localnames.	See "Default Provider / Consumer Artifact Types" in the *User Guide*	`SC-I`
`platform.cm.providers`	Artifact types defined as contract providers.	Comma-separated list of artifact localnames.	See "Default Provider / Consumer Artifact Types" in the *User Guide*	`SC-I`

Contract Management, continued

Property Name	Description	Value Type	Default	Scenario
`platform.cm.consumers.of.`*`provider`*	Artifact types defined as contract consumers for the specified provider artifact type.	Comma-separated list of artifact localnames.	None	`SC-I`
`platform.cm.providers.of.`*`consumer`*	Artifact types defined as contract providers for the specified consumer artifact type.	Comma-separated list of artifact localnames.	None	`SC-I`

Lifecycle

Property Name	Description	Value Type	Default	Scenario
`platform.lifecycle.notify.approvers`	Sets whether the approvers are automatically notified by e-mail containing approval request.	Boolean value.	*true*	`SC-I`
`platform.lifecycle.notify`	Turns off lifecycle notifications when set to false.	Boolean value.	*true*	`SC-I`
`platform.transaction.timeout`	Used to set transaction timeout for server	Integer value	1200	`SC-III`

Artifact Icons

Property Name	Description	Value Type	Default	Scenario
`platform.integration.discovery.bac.mappingConfiguration9`	Maps artifacts between Systinet and HP Business Service Management 9.x.	XML file which contains mapping from Systinet artifacts to BSM entities.	XML file mapping from Sytinet artifacts (SOAP Service, Business Service, Organization Unit) to BSM entities (webservice, business_ service, organization).	SC-III
`shared.ui.artifact-icons.`	Prefix of artifact icons configuration.	Prefix of artifact icons configuration.	Without default value.	`SC-I`

User Management

Property Name	Description	Value Type	Default	Scenario
`shared.um.scenarios.type`	Type of scenario for communication with LDAP server.	One of `single.ldap.[single or multiple].searchbase, , multiple.ldap.[single or multiple].searchbase` values.	*single.ldap.single.searchbase*	`SC-I`

User Management, continued

`shared.um.account.caseInsensitiveLoginName`	Sets whether login may be name case insensitive.	Boolean value	*true*	`SC-I`
`shared.um.account.backend.type`	Type of group backend.	Text value. One of `[ldap, database, external]`	Inserted in installation process.	`SC-IV`
`shared.um.account.backend.className`	Group backend class name.	Text representation of whole class name.	Inserted in installation process.	`SC-IV`
`shared.um.account.backend.enablemoreBackends`	By group backend allows more backends.	Boolean value.	*true*	`SC-IV`
`shared.um.group.backend.type`	Type of account backend.	Text value. One of `[ldap, database, external]`	Inserted in installation process.	`SC-IV`
`shared.um.group.backend.className`	Account backend class name.	Text representation of whole class name.	Inserted in installation process.	`SC-IV`
`shared.um.group.backend.enableMoreBackends`	By account backend allows more backends.	Boolean value.	*true*	`SC-IV`
`shared.um.account.domain.enabled.name`	Name of enabled domain.	Text representation name of enabled domain.	Not set.	`SC-I`
`shared.um.account.domain.enabled.dn`	Distinguished Name of enabled domain.	Text representation DN of enabled domain.	Not set.	`SC-I`
`shared.um.account.domain.disabled.name`	Name of disabled domain.	Text representation name of disabled domain.	Not set.	`SC-I`

User Management, continued

`shared.um.account.domain.disabled.dn`	Distinguished Name of disabled domain.	Text representation DN of disabled domain.	Not set.	`SC-I`
`shared.um.group.ldapName`	LDAP property name for group name.	Text value.	Inserted in installation process. (*cn*)	`SC-II`
`shared.um.group.ldapOwner`	LDAP property name for group owner.	Text value.	Inserted in installation process. (*owner*)	`SC-II`
`shared.um.group.ldapMember`	LDAP property name for group member.	Text value.	Inserted in installation process. (*uniqueMember*)	`SC-II`
`shared.um.group.ldapDescription`	LDAP property name for group description.	Text value.	Inserted in installation process. (*description*)	`SC-II`
`shared.um.account.ldapLoginName`	LDAP property name for login name.	Text value.	Inserted in installation process. (*uid*)	`SC-II`
`shared.um.account.ldapFullName`	LDAP property name for full name.	Text value.	Inserted in installation process. (*cn*)	`SC-II`
`shared.um.account.ldapEmail`	LDAP property name for email.	Text value.	Inserted in installation process. (*mail*)	`SC-II`
`shared.um.account.ldapDescription`	LDAP property name for account description.	Text value.	Inserted in installation process. (*description*)	`SC-II`

User Management, continued

shared.um.account.ldapLanguageCode	LDAP property name for language code.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapBusinessName	LDAP property name for business name.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapPhone	LDAP property name for phone.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapAlternatePhone	LDAP property name for alternate phone.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapAddress	LDAP property name for address.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapCity	LDAP property name for city.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapCountry	LDAP property name for country.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapBlocked	LDAP property name for index of blocking.	Text value.	Inserted in installation process.	SC-II
shared.um.account.ldapZip	LDAP property name for zip code.	Text value.	Inserted in installation process.	SC-II

User Management, continued

`shared.um.uddi.ldap.searchfilter.group`	LDAP search filter for group.	Text value.	Inserted in installation process. (*objectClass=groupofuniquenames*)	`SC-IV`
`shared.um.uddi.ldap.searchbase.group`	LDAP search base for group.	Text value.	Inserted in installation process. (*ou=groups,ou=big, dc=example,dc=com*)	`SC-IV`
`shared.um.uddi.ldap.searchscope.group`	These property sets search scope for users and group entries within an LDAP : 0 = object scope (useless in this case), 1 = one level (flat) scope, 2 = subtree scope.	Integer value.	Inserted in installation process. (*2*)	`SC-IV`
`shared.um.uddi.ldap.searchMaxResults.group`	LDAP search max results for user.	Integer value.	Inserted in installation process. (*10000*)	`SC-IV`
`shared.um.uddi.ldap.searchfilter.user`	LDAP search filter for user.	Text value.	Inserted in installation process. (*objectClass=person*)	`SC-IV`
`shared.um.uddi.ldap.searchbase.user`	LDAP search base for user.	Text value.	Inserted in installation process. (*ou=people,ou=big, dc=example,dc=com*)	`SC-IV`

User Management, continued

`shared.um.uddi.ldap.searchscope.user`	These property sets search scope for users and group entries within an LDAP : 0 = object scope (useless in this case), 1 = one level (flat) scope, 2 = subtree scope.	Integer value.	Inserted in installation process. (*2*)	`SC-IV`
`shared.um.uddi.ldap.searchMaxResults.user`	LDAP search max results for user.	Integer value.	Inserted in installation process. (*10000*)	`SC-IV`
`shared.um.uddi.ldap.allowBlankPassword`	LDAP sign for allowing blank password.	Boolean value.	Inserted in installation process. (*false*)	`SC-IV`
`shared.um.java.naming.provider.url`	LDAP url.	URL.	Inserted in installation process. (*ldap://ldap.example.com:389*)	`SC-IV`
`shared.um.java.naming.factory.initial`	LDAP factory.	Full LDAP factory class name.	Inserted in installation process. (*com.sun.jndi.ldap.LdapCtxFactory*)	`SC-IV`
`shared.um.java.naming.security.principal`	LDAP principal.	Text value.	Inserted in installation process.	`SC-IV`
`shared.um.java.naming.security.credentials`	LDAP credentials.	Text value.	Inserted in installation process.	`SC-IV`

User Management, continued

`shared.um.java.naming.security.authentication`	LDAP authentication type. `[none, simple, sasl_mech]` where `sasl_mech` is a space-separated list of SASL mechanism names.	Text value.	Inserted in installation process.	`SC-IV`
`shared.um.ldap.connect.timeout`	Timeout for connection to LDAP.	Integer value.	*0*	`SC-I`
`shared.um.ldap.connect.pool`	Enabling connection pooling.	Boolean value.	*false*	`SC-I`
`shared.um.ldap.connect.pool.authentication`	Connection pooling authentication value.	Text value.	Not used.	`SC-I`
`shared.um.ldap.connect.pool.debug`	Connection pooling debug switch.	Boolean value.	Not used. (*false*)	`SC-I`
`shared.um.ldap.connect.pool.initsize`	Connection pooling initial size.	Integer value.	Not used. (*1*)	`SC-I`
`shared.um.ldap.connect.pool.maxsize`	Connection pooling max size.	Integer value.	Not used. (*no max size*)	`SC-I`
`shared.um.ldap.connect.pool.protocol`	Connection pooling protocol.	Text value.	Not used. (*plain*)	`SC-I`
`shared.um.ldap.connect.pool.timeout`	Connection pooling connection timeout.	Integer value.	Not used. (*no timeout*)	`SC-I`

Uncategorized

Property Name	Description	Value Type	Default	Scenario
`shared.smtp.host`	SMTP host name.	Text representation of host name.	*localhost*	`SC-II`, `SC-III`
`platform.smtp.host`	SMTP host name.	Text representation of host name.	*localhost*	`SC-II`, `SC-III`
`shared.smtp.port`	SMTP port.	Integer value.	*25*	`SC-II`, `SC-III`
`platform.smtp.port`	SMTP port.	Integer value.	*25*	`SC-II`, `SC-III`
`shared.smtp.auth`	SMTP authentication switch.	Boolean value.	*false*	`SC-II`, `SC-III`
`platform.smtp.auth`	SMTP authentication switch.	Boolean value.	*false*	`SC-II`, `SC-III`
`shared.smtp.auth.username`	SMTP user name name.	Text value.	Not used.	`SC-II`, `SC-III`
`platform.smtp.auth.user`	SMTP user name name.	Text value.	Not used.	`SC-II`, `SC-III`
`shared.smtp.auth.password`	SMTP user password.	Text value.	Not used.	`SC-II`, `SC-III`
`platform.smtp.auth.password`	SMTP user password.	Text value.	Not used.	`SC-II`, `SC-III`
`shared.http.urlbase`	HTTP URL base.	URL.	Inserted in installation process.	`SC-IV`
`shared.http.port`	HTTP port.	Integer value.	Inserted in installation process. (*8080*)	`SC-IV`
`shared.https.urlbase`	HTTPS URL base.	URL.	Inserted in installation process.	`SC-IV`

Uncategorized, continued

`shared.https.port`	HTTPS port.	Integer value.	Inserted in installation process. (*8443*)	`SC-IV`
`shared.https.use`	Switch for use HTTPS.	Boolean value.	*false*	`SC-IV`
`platform.webui.max.branch.count`	Maximum number of concurrent windows.	Integer value.	4	`SC-III`
`platform.webui.max.branch.depth`	Page browsing history limit.	Integer value.	10 (effectively 5 pages)	`SC-III`
`shared.db.fulltextsearch.appendpercentage`	Determines whether a % is appended to search terms	TRUE or FALSE	TRUE	`SC-III`
`platform.webui.max.upload.size`	Determines maximum size of a repository archive file for import.	Unlimited.	-1	`SC-I`
`shared.ui.operation.color.restMethods`	Defines the foreground and background colors.	XML file	XML file defines the foreground and background colors of methods in HTTP Requests table of REST overview page.	`SC-I`

Uncategorized, continued

`platform.ui.breadcrumb.configuration`	Breadcrumb path definition	XML file	Breadcrumb path definition of interested artifact types: Application, REST Service, SOAP service, HTTP Request, HTTP Response, SOAP operation, HTTP Endpoint, SOAP Endpoint	`SC-I`

RGIF Discovery

Property Name	Description	Value Type	Default	Scenario
`platform.rgifdiscovery.local.changes.mode`	Specifies whether to keep the local changes. 2 possible values: • Keep • Overwrite	Text value	Keep	`SC-I`

RGIF Discovery, continued

`platform.rgifdiscovery.synchronization.mode`	Specifies the rgif synchronization mode. 4 possible values: • All • Add New • Remove Obsolete • Update Changes	Text value	ALL	`SC-I`
`platform.rgifdiscovery.wsdls.decomposition`	Specifies whether to decompose WSDL of runtime services. 3 possible values: • None • Implementations • Services	Text value	None	`SC-I`

How to Manage Artifact Form Validation

The Administrator can assign policies to validate artifacts when a user creates or edits them. For example, to ensure that keywords are set when a user creates a service.

To Assign Artifact Validation Policies:

1. In the Administration tab->Administration menu, click **Configuration** to open the Configuration page.

2. Select the **Artifact Validation** tab to view the list of artifact types.

   > **Note:** Numbers next to artifact type names indicate validation policies in use.

3. Select the artifact type that you want to validate for creation and edit. The Validated By table refreshes to show the policies used to validate the selected artifact type.

4. Do one of the following:

   - **To Add Policies:**

     i. Click **Add Policy** to open the Add Policy dialog.

     ii. Use the search criteria to locate the technical policies you want to use.

     iii. Select the policies to use and click **Select** to add the policies to the Validate By table.

   - **To Remove Policies:**

     i. Select the policies to remove.

     ii. Click **Remove Selected** and confirm your decision to remove the selected policies from the Validated By table.

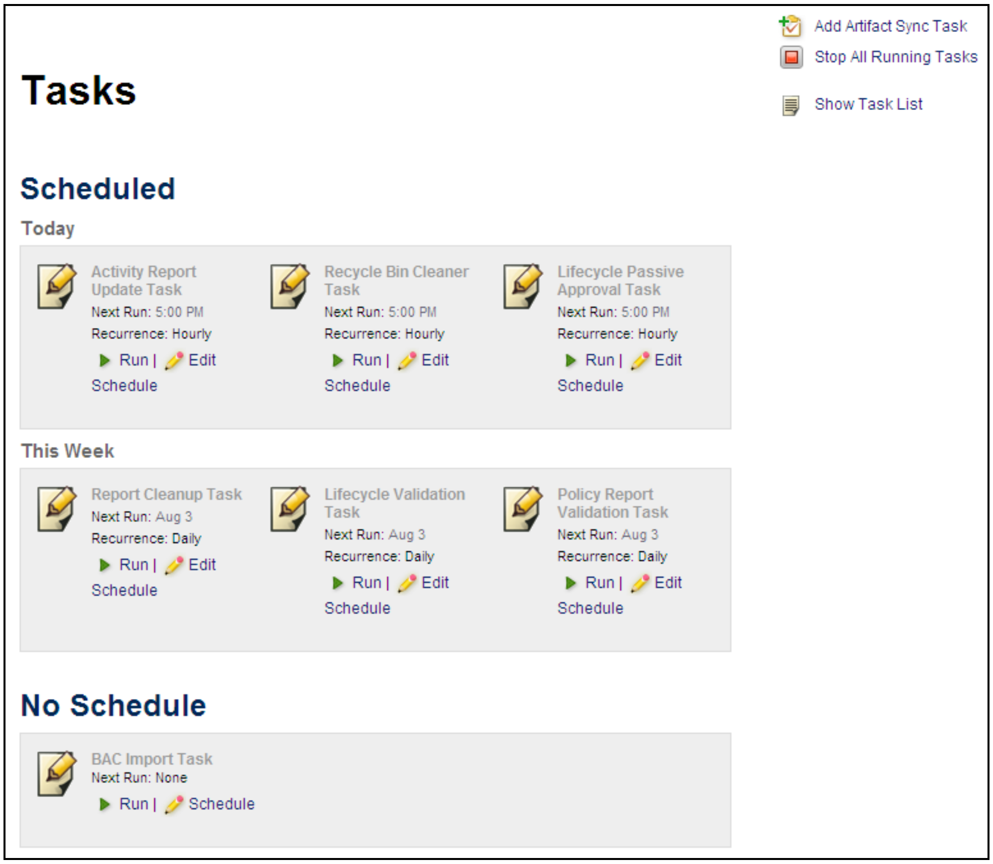
You can extend form validation to also validate data attachments to artifacts. Set property `platform.autovalidation.validateData` to `true`. For details, see "How to Manage System Settings" on page 81.

Chapter 10: Administration Task Management

Systinet includes a number of default administration tasks to help administrators to manage their deployment and the data it contains. In addition, the administrator can configure additional tasks to perform change management or custom tasks included in an extension.

Access Task Management from the Administration tab. In the Administration menu, click **Tasks** to open the Tasks page.

Screenshot: Tasks Page



Add Artifact Sync Task

Stop All Running Tasks

Show Task List

Tasks

Scheduled

Today

Activity Report Update Task
Next Run: 5:00 PM
Recurrence: Hourly
Run | Edit Schedule

Recycle Bin Cleaner Task
Next Run: 5:00 PM
Recurrence: Hourly
Run | Edit Schedule

Lifecycle Passive Approval Task
Next Run: 5:00 PM
Recurrence: Hourly
Run | Edit Schedule

This Week

Report Cleanup Task
Next Run: Aug 3
Recurrence: Daily
Run | Edit Schedule

Lifecycle Validation Task
Next Run: Aug 3
Recurrence: Daily
Run | Edit Schedule

Policy Report Validation Task
Next Run: Aug 3
Recurrence: Daily
Run | Edit Schedule

No Schedule

BAC Import Task
Next Run: None
Run | Schedule

The initial view of the Administrator Task page provides a simplified grid interface for dealing with the administration of Daily, Weekly, and unscheduled tasks. Any running tasks will also be visible with a options to either stop them independently or to stop all running tasks.

Systinet provides the following default administration tasks:

- **Activity Report Update**

  Updates the artifact activity data used to generate the Activity Report shown in the Reports tab Homepage.

- **Lifecycle Passive Approval**

  Approves Lifecycle stages that have been set to automatic approval after a set number of days.

- **Lifecycle Validation**

  Performs a validation of all artifacts in governance against the policies that apply to their current lifecycle stage.

- **Policy Report Validation**

  Performs an update of any policy reports. For more details, see "Policy Reports" in the *User Guide*.

- **Recycle Bin Cleaner**

  Permanently removes deleted artifacts from the recycle bin as the defined schedule.

- **Report Cleanup**

  Removes old reports and events based on their age against properties set in the Configuration page Report Cleanup tab. For details, see "Configuration Management" on page 76.

The administrator can execute tasks immediately or schedule them to run on a periodic basis. For details, see "How to Run Tasks" below and "How to Schedule Tasks" below.

In addition to the default tasks, the administrator can add change management tasks and custom tasks. For details, see "How to Add Change Management Tasks" on the next page and "How to Add Custom Tasks" on page 107.

How to Run Tasks

You may want to immediately execute a task. If you have sufficient permissions, there is a **Run** option in the initial Tasks overview as well as in the detail view of the task.

To Manually Execute Tasks:

- Click **Run** in either the Task overview or in the Task detail page.

How to Schedule Tasks

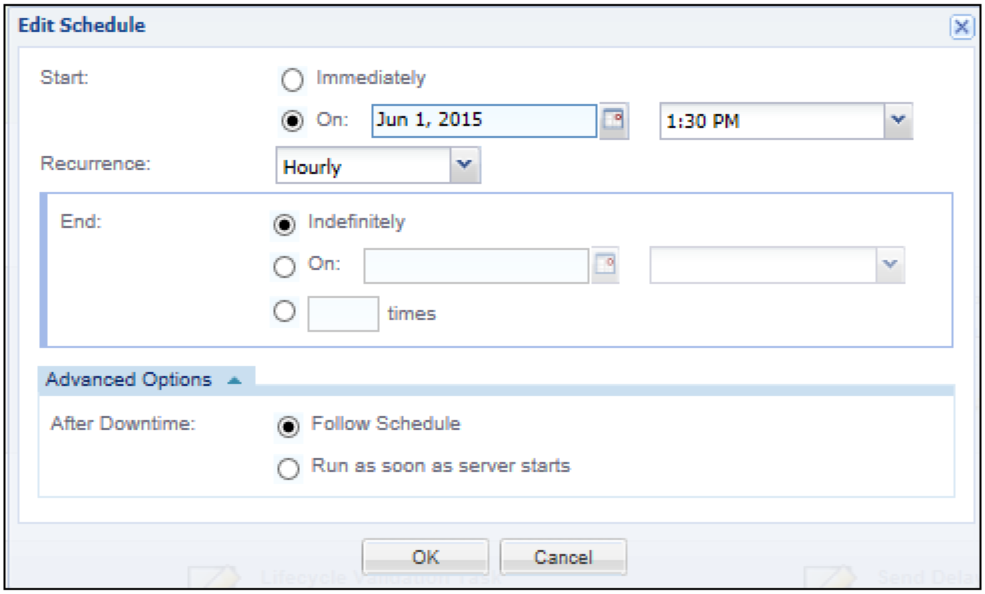
Systinet enables you to execute tasks on a timed or periodic basis.

> **Caution:** Systinet converts and stores any input time to GMT. If you import scheduled tasks from a data image, review the scheduling to ensure that the tasks execute at the local time that you require.

To Schedule Tasks:

- Do the following:
  - In the task window or detail view of the task click **Edit Schedule** to open the Edit Schedule dialog.

    Screenshot: Edit Schedule Dialog

    Start: ○ Immediately
    ● On: Jun 1, 2015 1:30 PM

    Recurrence: Hourly

    End: ● Indefinitely
    ○ On:
    ○ times

    After Downtime: ● Follow Schedule
    ○ Run as soon as server starts

  - Select a start date and start time for the task
  - Set the frequency that the task will run
  - Either set the task to run Indefinitely, set a finish date and time or enter a value representing the number of times the task should run
  - Advanced Options allow you to select whether the schedule should be followed during downtime or to wait until Systinet start running.
  - Click **OK** to set the schedule for the task

How to Add Change Management Tasks

Sync Tasks enable the administrator to synchronize artifacts in the repository with those imported from an external source

To Create Sync Tasks:

1. Click **Add Artifact Sync Task** to open the dialog.
2. Add a name and optional description for the task.

3. Artifacts can be selected through using either standard search methods or they can be selected from a filtered list.

   Click **Save** .

4. Clicking **Edit Schedule** will allow you to set a schedule for the running of the new task or the task can be run immediately by clicking **Run**.

5. The task may be edited by clicking on **Edit**.

How to Add Custom Tasks

If Systinet includes an extension which contains custom task implementations, you can add and schedule them in the Tasks page.

To Create Custom Tasks:

1. Click **Add Custom Task** to open the dialog.

2. Select a Task Implementation and add a name for the task.

3. Task Parameters are defined and variable according to how the task component was composed.

   Click **Save** .

4. Clicking **Edit Schedule** will allow you to set a schedule for the running of the new task or the task can be run immediately by clicking **Run**.

5. The task may be edited by clicking on **Edit**.

Chapter 11: Repository Data Diagnostic Tool

The Repository Data Diagnostic Tool (DDT) can be introduced as a scheduled task that generates a report. While running, DDT collects test suites, which are contained in script artifacts. The test suite provides broad data consistency checking of SQL, DQL, and technical policies.

The tool performs the following tasks:

- Ensures all relations are functional with each other and have no dead endpoints.
- Discovers orphaned artifacts and ensures that there are no data artifacts missing incoming or outgoing relations (for example, orphaned endpoints).
- Reports on artifacts in non-existing life cycle stages. The LFE Data Validity script checks artifacts in both non-existing lifecycle processes and lifecycle stages. The script also checks lifecycle stage artifacts with invalid/error status and checks lifecycle stage artifacts with incorrect number approval.
- Determines that taxonomy values and names are correct. The Taxonomy Usage Validity script artifact checks for valid taxonomy ID format, duplicate taxonomy ID, and duplicated taxonomy categories. The script also reviews Artifact overall for invalid taxonomy (i.e., if artifact uses a non-existing category or a non-existing taxonomy ID).
- Certify that mandatory/required fields contain data and/or correct existing data.

DDT supports a paging option for SQL and DQL queries. The default page size is 50 and can be changed in the Administration module under the Configuration > System Settings tab: property named `platform.diagnosticTool.query.pageSize`. Paging is disabled if the value = -1.

- "How to Run or Schedule a DDT Task" below
- "How to View a DDT Report" below

How to Run or Schedule a DDT Task

From Administration module, select **Task** and click on **Repository Data Diagnostic Task**.

Using the wizard, you can run or schedule a DDT task.

See "How to Schedule Tasks" on page 105 for more information.

How to View a DDT Report

Click **See report for more details** or **Finished** after running the DDT task.

Chapter 12: UI Customization

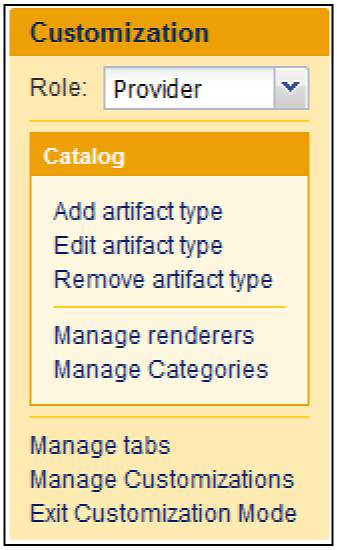
The primary purpose of the role-based UI is to present users with only the information and functionality relevant to their roles. As the nature of these roles varies in different organizations, the UI displayed for each role in Systinet is customizable.

Each element in a role-based UI is customizable in the same way.

To customize a role-based UI:

In the Administration tab, click **Customize** to open the customization menu and select the ***ROLENAME***. from the dropdown menu.

Screenshot: Customization Menu



The UI for that role opens in customization mode.

You can navigate the role-based UI in customization mode in the same way as a user in that role. Each customizable UI component has its own **Customize** link enabling you to modify it.

Customize individual UI elements as required referring to the following sections which are organized according to the pages where the elements appear. UI Customizations provide the most commonly required customizations and automate the process, but if you need more detailed customization you may use the XML Customizations.

- UI Customizations:
  - "Adding and Editing Artifacts in the UI" on page 111
  - "How to Customize Renderers" on page 133.
  - "How to Customize Categories" on page 135
  - "How to Create Role-Based Tabs" on page 137

- XML Customizations:
  - "How to Customize Menus" on page 114
  - "How to Customize Homepages" on page 116
  - "How to Customize Browse Pages" on page 118
  - "How to Customize Create / Edit Artifact Pages" on page 125
  - "How to Customize Artifact Detail Pages" on page 128
  - "Adding Related Artifacts to the UI" on page 113
  - "Adding New Properties to the UI" on page 113

> **Tip:** For ease of customization, HP recommends the copying of existing customizations and pasting them into new customizations. For example, most browse pages are similar, and only the artifact types differ.

> **Note:** This is not a comprehensive list of customizable UI elements but the customizations described should enable you to customize other pages and components. Refer to "Role Customization Reference" on page 145 for full details of each customizable UI element.

Click **Manage Customizations** to open the Manage Customizations page displaying a list of prepared customizations.

For more details about managing your customizations, see "How to Manage UI Customizations" on page 136.

Select the customizations to release, and select **Release**.

The selected customizations are released to the server and all users in the role use the customized UI the next time they sign in to Systinet .

Customizations can also be imported or deleted from this page.

Clicking **Exit Customization Mode** returns you to the normal Catalog page.

UI Customization Use Cases

Apart from changing the layout and content of the UI, the main use case for UI customization is to incorporate your Service Definition Model customizations into the UI.

> **Note:** WYSIWYG customizations are not supported for Person artifacts. Use XML customizations if required.

This section describes the main customization use cases in the following sections:

- "Adding and Editing Artifacts in the UI" below
- "Adding New Properties to the UI" on page 113
- "Adding Related Artifacts to the UI" on page 113

Adding and Editing Artifacts in the UI

After you add a new artifact to your SDM, you may want it to be available to users in role-based UIs.

To add an artifact to the UI:

1. In Manage Customizations mode, select the user role that requires the artifact and click **Add artifact type** .

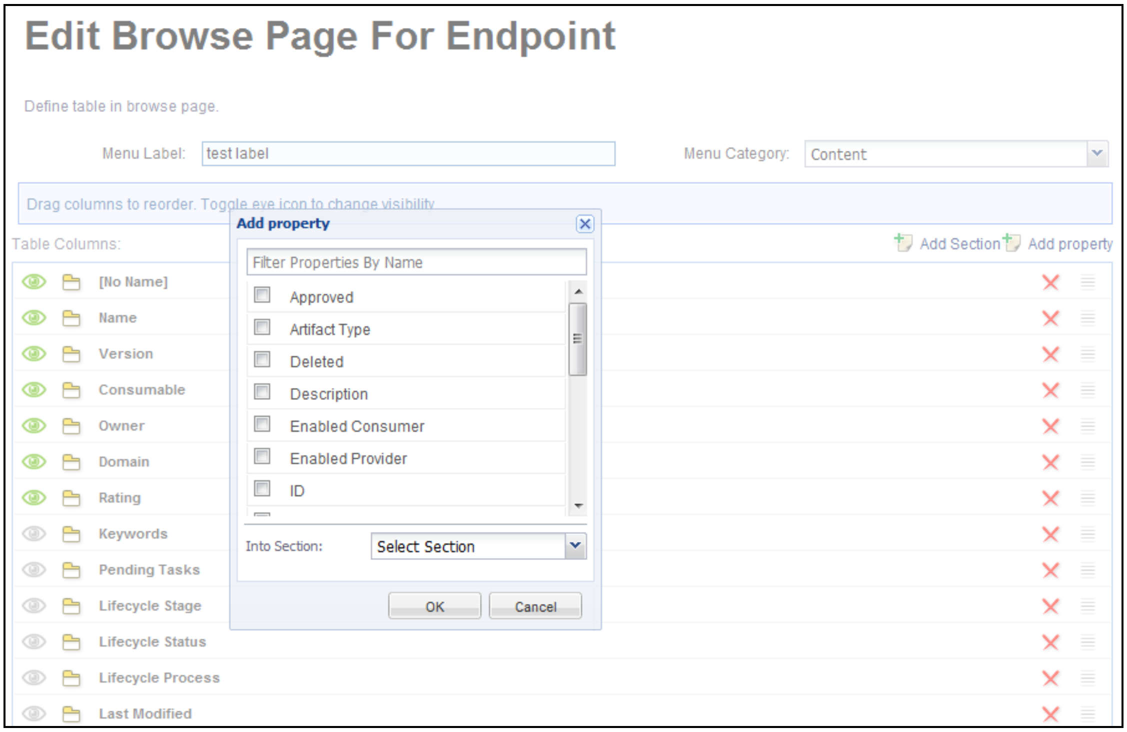
2. Select the required artifact type from the tree view of artifact types.

3. Provide Menu labels for singular and plural artifacts and a Menu category.

   From the table, Sections and properties can be added, column visibility can be switched on and off and column location may be changed by dragging and dropping them as required.

   Columns may also be deleted.

   Screenshot: Edit Browse Page



Edit Browse Page For Endpoint

Define table in browse page.

Menu Label: test label

Menu Category: Content

Drag columns to reorder. Toggle eye icon to change visibility

Into Section: Select Section

Sections and properties defined here will be visible on the Browse Page.

Click **Next** to continue

4. The next screen allows you to select which tabs should be visible for the new artifact (View Page) and how they should be arranged as well as the ability to add new sections and properties to new or existing tabs.

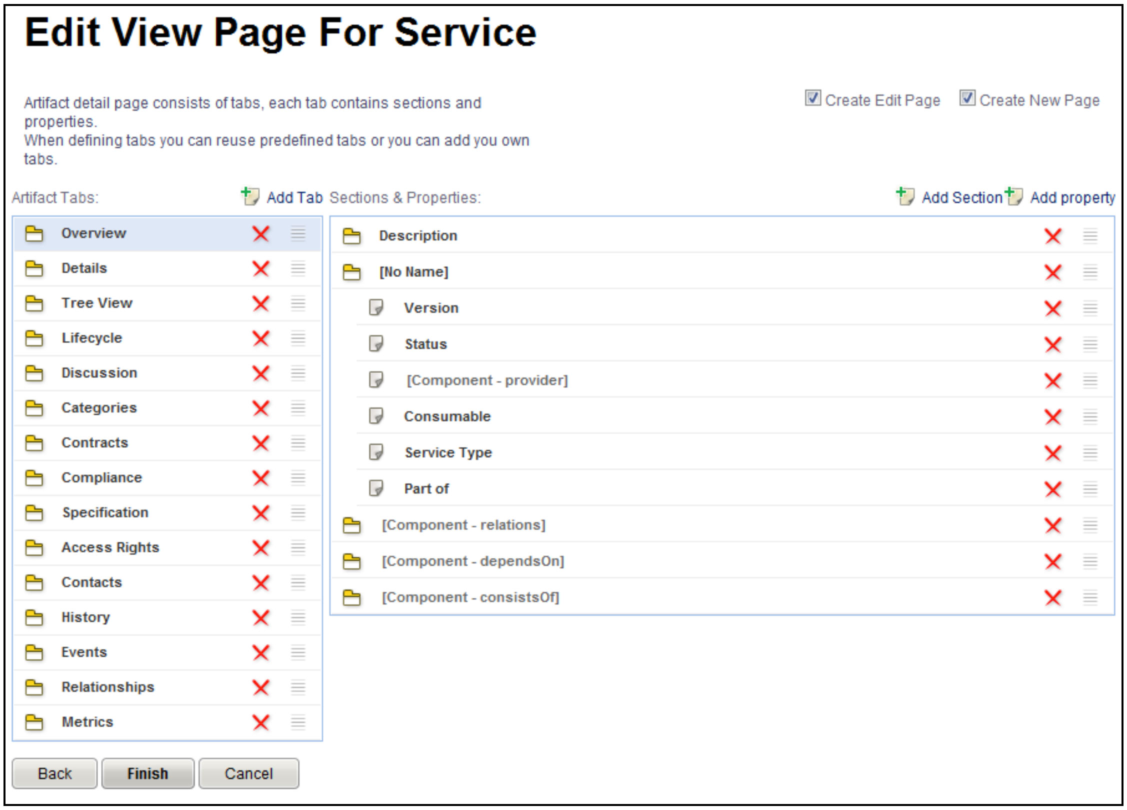
   There are three tab types:

   System - Lifecycle, Details, Discussion, etc.. Content is not customizable

   Property - Artifact properties and is customizable

   Categories - Categories with ability to add and remove

   Screenshot: Edit View Page



Edit View Page For Service

Artifact detail page consists of tabs, each tab contains sections and properties.
When defining tabs you can reuse predefined tabs or you can add you own tabs.

☑ Create Edit Page ☑ Create New Page

Artifact Tabs: Add Tab

- Overview
- Details
- Tree View
- Lifecycle
- Discussion
- Categories
- Contracts
- Compliance
- Specification
- Access Rights
- Contacts
- History
- Events
- Relationships
- Metrics

Sections & Properties: Add Section Add property

- Description
- [No Name]
  - Version
  - Status
  - [Component - provider]
  - Consumable
  - Service Type
  - Part of
- [Component - relations]
- [Component - dependsOn]
- [Component - consistsOf]

Create Edit Page and **Create New Page** options automatically create the Edit or New pages if selected.

If different renderers are available they will be shown in a dropdown list beside the property item. See "Renderers" on page 114 for more details.

Click **Finish** to add the Artifact.

To edit an existing artifact:

1. In Manage Customizations mode, select the user role that requires the artifact edited and click **Edit artifact type** .
2. Select the required artifact type from the tree view of artifact types.
3. Edit the artifact in the same manner as steps 3 and 4 when adding an artifact

To remove an artifact from the UI:

1. In Manage Customizations mode, select the user role that requires the artifact edited and click **Remove artifact type** .
2. Select the artifacts for removal by clicking the associated select boxes.
3. Click **Remove**.

Note: Removing an artifact deletes the artifact from view immediately. It does not require a release of the customization.

Adding New Properties to the UI

After you add a simple or taxonomic property to an artifact type, you may want it to be visible to users in a role-based UI.

To add a property to the UI:

1. To enable users to view the value of the property for artifact instances, customize the Artifact Detail page for the artifact type to add the simple or taxonomic property to the artifact details.

   For details, see "How to Customize Artifact Detail Pages" on page 128.

2. To enable users to add or edit a property value to new or existing artifacts, customize the Create and Edit pages for the artifact type.

   For details, see "How to Customize Create / Edit Artifact Pages" on page 125.

3. To enable users to sort or filter the artifact type based on the property, add the property to tables showing the artifact type or add the property to search criteria for the artifact type.

   For details, see "How to Customize Related Artifact Tables" on page 132 and "How to Customize Browse Page Search Criteria" on page 122.

Adding Related Artifacts to the UI

After you add a relationship property to an artifact type you may want to show a table of the related artifacts in the artifact details tab.

To add a relationship to the UI:

- In the artifact details tab or artifact create and edit pages, add a table of related artifacts to the artifact details section.

  For details, see "How to Customize Artifact Detail Pages" on page 128” and the example in "How to Customize Create / Edit Artifact Pages" on page 125”.

Renderers

Artifacts can be and typically are connected with other artifacts via relationships. For example, there is an implementation relationship between Service and SOAP Service artifacts. InSystinetyou can view a table of SOAP Services in the detail tab of the Service artifact detail page or you can view a table of documentation artifacts on the details tab of service artifact detail page.

A Renderer defines a table of related artifacts. For our example, we'll use the table of SOAP Services or Documentation artifacts in the Service detail page. When you customize the UI, you can define not only whether a relationship property is displayed but you can also define the table of related artifacts. Imagine that you want to customize the table of Documentation artifacts in Service detail page. You can create a new renderer and then change the service detail page definition or you change the existing renderer which means that your change is displayed in all places where that renderer is used. A renderer would also be necessary if you want to add custom artifact types with relations to other custom artifact types as in this case, the new artifact types would not have any default relationship.

See "Relationships" in the *User Guide* and "How to Customize Renderers" on page 133 for more information.

How to Customize Menus

Each page in Systinet contains the menu on the left, and a page layout on the right.

The menu is organized into groups and items. Items are usually navigation links to other pages or specific functionality, or a component which performs some function (for example, Recent Documents).

To customize a menu:

1. Navigate to any page in customization mode for the role-based UI that you want to modify.

2. Click the **Customize** link above the menu to open the XML representation of the menu.

   The menu forms the following hierarchy of elements:

   - <navigation>

     The holding element for the menu.

- <group>

  Each group consists of a set of tasks and components with a heading.

  - <task>

    A navigation link to another page.

    Tasks in menus may be one of the following:

    - /analyst/browseArtifact
    - /analyst/newArtifact
    - /partner/browseArtifact
    - /artifact/*UUID*

    For details, see “Reusable Tasks”.

  - <component>

    A more complex piece of functionality.

3. Modify the menu as required,

   For more details, see "Menu Customization Reference" on page 142

4. Click **Save** to save your menu customization locally.

The following example describes how to add new links to service level objective (SLO) pages to the Business Analyst UI menu.

To add a new artifact type to the menu:

1. Click **Customize** to open the XML representation of the menu.

2. To add a link to the SLOs browse page, in the `view` group, add the following `task` element after the `contracts` task element:

```
<task id="analyst.view.slo" taskLocation="/analyst/browseArtifact"
label="SLOs">
  <parameter name="artifactType">sloArtifact</parameter>
</task>
```

3. To add a link to the new SLO page, click **Customize** above the Artifacts menu, than add the following code after the project task element:

```
<artifactType default="true" id="sloArtifact"/>
```

4. Click **Save** and wait a few moments for the page to refresh.

The new custom menu should now contain **SLO** links in both the View and New groups of the menu.

> **Note:** Clicking the SLOs view link opens an empty page as the page for browsing SLOs is not defined by default in the Analyst UI. "How to Customize Browse Pages" on page 118 describes the creation of a Browse SLOs page.

How to Customize Homepages

Each role-based UI opens with a homepage for that role when a user logs in.

You can customize the following elements of the homepage:

- Customize the page layout using the **Customize** link at the top right of the page.

  For details, see "How to Customize Homepage Layouts" below.

- Customize HTML area Useful Links, using the **Edit** link in the component.

  For details, see "How to Customize HTML Areas" on the next page.

- Customize Search using the **Customize** link in the Search component.

  For details, see "How to Customize Homepage Search" on the next page.

- Customize Latest Events by clicking on the Customize link in the Latest Events Component

  For details, see "How to Use Events" in the *User Guide*.

How to Customize Homepage Layouts

You can customize the order and placement of components on the homepage as well as the value of some of the components (if applicable).

To customize the homepage layout:

1. In the customization view of the role specific UI, navigate to the homepage, and click **Customize** in the top right of the page.

   An XML editor window opens displaying the XML extract for the page.

   The page layout forms the following hierarchy of elements:

   - <columns>

    The holding element for a page layout.

    - <column>

      Contains the content of each column, usually a set of components organized into one or two columns.

      - <component>

        A more complex piece of functionality, such as the Search component.

2. Modify the XML extract as required.

   For more details, see "Page Layout Customization Reference" on page 144”.

   > **Tip:** HP recommends only using this customization to move components on the page or to remove them from the page.

3. Click **Save** to save your changes locally.

How to Customize HTML Areas

On the homepage, **Useful Links** is a customizable HTML area.

To customize an HTML area:

1. In the customization view of the role specific UI, navigate to the homepage.

2. Click **Edit** for the HTML area you want to customize. The rich-text HTML editor for the component opens.

3. Edit the rich text HTML as required.

   You can edit rich text using the Editor toolbar buttons, or switch to source HTML editing with the button.

   For details of allowed source HTML content, see "HTML Areas Customization Reference" on page 168.

4. Click **Save** to save your changes locally.

How to Customize Homepage Search

You can customize the Search component on the homepage.

To customize a homepage search component:

1. In the customization view of the role-specific UI, navigate to the homepage.
2. Click **Customize** in the Search component. The XML editor for homepage search opens.

   The homepage search forms the following hierarchy of elements:

   - <suggestConfig>

     The containing element for the search component and defines the dynamic results behaviour.

     - <datasource>

       Defines the query for the search.

     - <search>

       Defines the criteria for the search specifying which properties of the artifact types specified by the query are searched.

     - <table>

       Defines the layout of the dynamic results table

     - <actionComponent>

       Determines the navigation behaviour when you click an artifact in the dynamic results table.

3. Modify the XML extract as required.

   For more details, see "Search Component Customization Reference" on page 164”.

4. Click **Save** to save your changes locally.

How to Customize Keywords

By default, 20 keywords are displayed. This can be modified by clicking the **Customize** button while in Customize mode and changing the maximumTagsDisplayed value in the XML code.

Note: Changing this value to 0 displays all tagged keywords.

How to Customize Browse Pages

The links in the Artifact part of the menu open Browse Artifact pages for the particular artifact type. Browse pages consist of multiple sets of filters, search criteria, and results tables and so provide multiple ways to browse a particular artifact type.

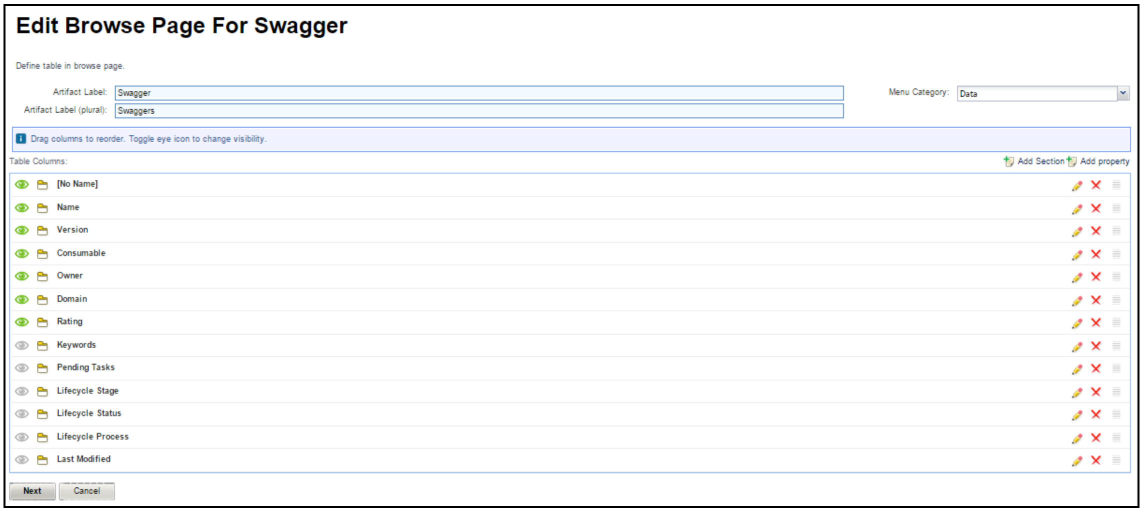
Each filter (for example, **All** or **Favorites**) represents a particular query on the artifact type and links to a particular set of search criteria and a results table.

> **Caution:** To enable browse pages for an artifact type, make the artifact type a primary artifact in the tab configuration. For details, see “Tab Configuration”.

The default browse page consists of the following customizable elements:

- The **Edit Page** button at the top-right of the page enables you to customize the table layout by adding or subtracting columns, rearranging column location, adding sections and adding properties to those sections.

  Screenshot: Edit Browse Page

  Edit Browse Page For Swagger

  Define table in browse page.

  Artifact Label: Swagger
  Artifact Label (plural): Swaggers
  Menu Category: Data

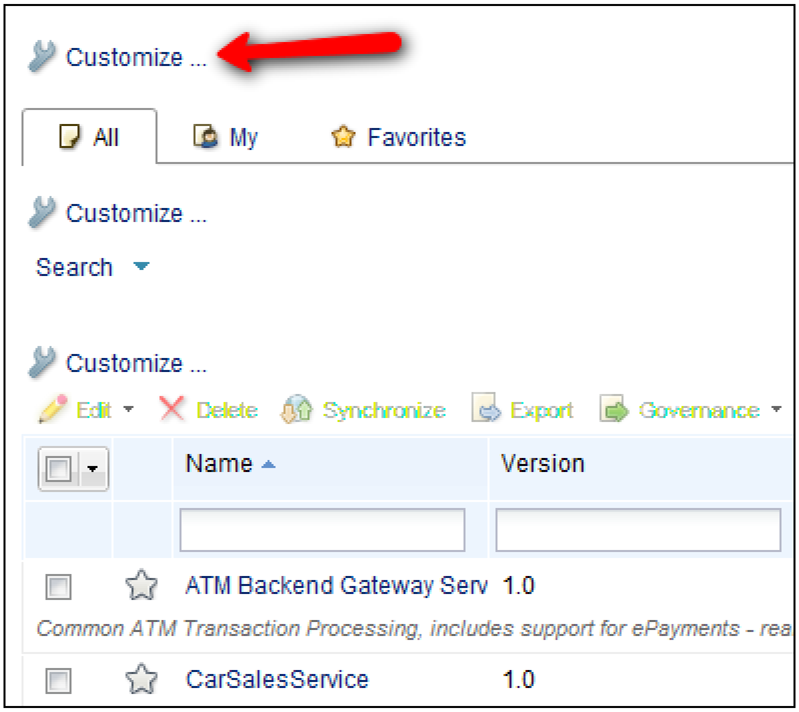
  Drag columns to reorder. Toggle eye icon to change visibility.

  Table Columns:

  [No Name]
  Name
  Version
  Consumable
  Owner
  Domain
  Rating
  Keywords
  Pending Tasks
  Lifecycle Stage
  Lifecycle Status
  Lifecycle Process
  Last Modified

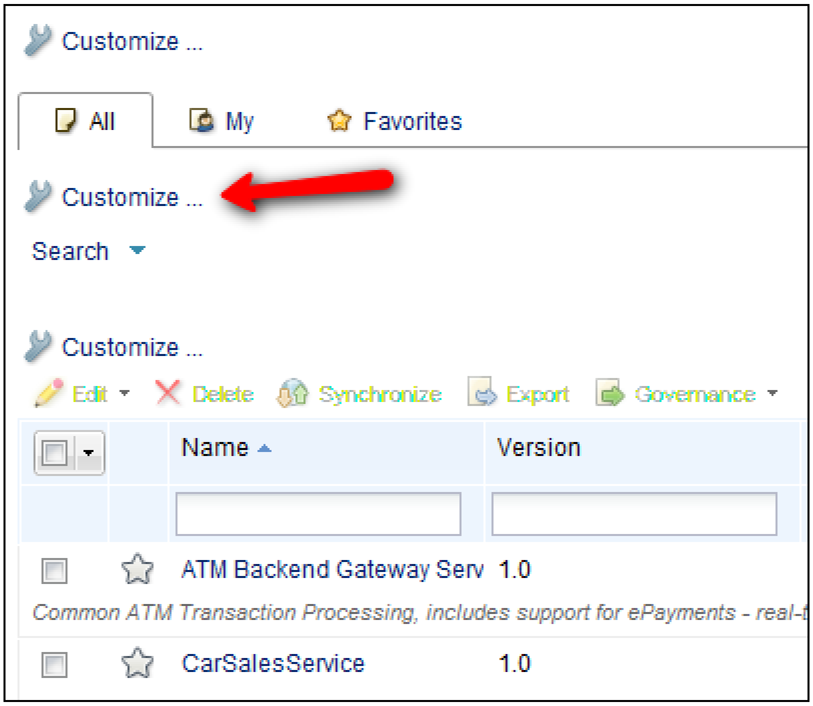
- The customize link above the tabs enables you to customize the page filters.

Screenshot: Customize Filter



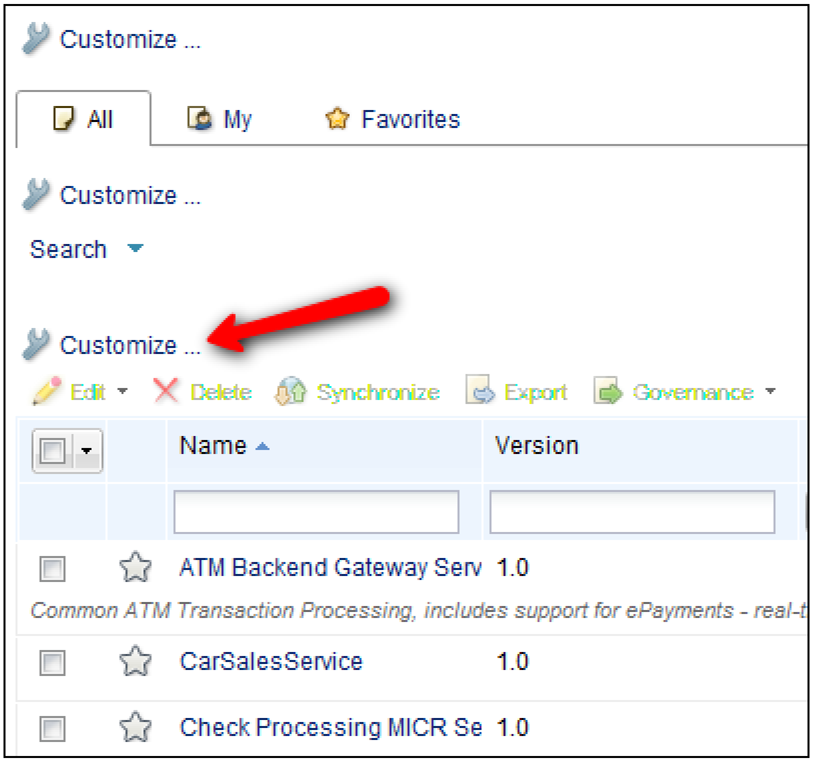
- The customize link in the search box enables you to customize search criteria.

Screenshot: Customize Search



- The customize link in the artifact table enables you to customize the XML for the table of artifacts containing the search results.

Screenshot: Customize XML



For details of these customizations, see the following sections:

- "How to Customize Browse Page Filters" below
- "How to Customize Browse Page Search Criteria" on the next page
- "How to Customize Browse Page Tables" on page 123

How to Customize Browse Page Filters

The content of the table in browse pages and the criteria used by the search component are linked to search filters. Each filter is a query on the artifact type.

Each filter is associated with a corresponding set of search criteria and a table. The following example sets two filters for the SLOs browse page, All SLOs, and Favorite SLOs:

To create a set of browse page filters:

1. Open the **SLOs** browse page, as described in “Customizing Browse Artifact Pages”.

2. Click **Customize** above the filters to open the XML representation of the page filters.

3. Replace the default content with the following XML extract:

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
      xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="cust artifactView.xsd">
  <queries>
    <item icon="All" id="allServices" label="All">
      <datasource>
           <query> select from sloArtifact s </query>
      </datasource>
      <parameter name="criteria">${customizationId}.criteria</parameter>
      <parameter name="table">${customizationId}.results</parameter>
    </item>
    <item icon="Favorites" id="favoriteServices" label="Favorite"
selected="true">
      <datasource>
        <query> select from sloArtifact s where s._isFavorite = 1 </query>
      </datasource>
      <parameter name="criteria">${customizationId}.criteria</parameter>
      <parameter name="table">${customizationId}.results</parameter>
    </item>
  </queries>
</customization>
```

   Each filter consists of a datasource containing a DQL query. For more details about DQL queries, see "Using DQL" in the *Developer Guide*.

   For more details, see "Query Selector Customization Reference" on page 159.

4. Click **Save** to create your SLO search filters.

To complete the application of the filters modify the search component and table for each filter.

For details of these customizations, see the following sections:

- "How to Customize Browse Page Search Criteria" below

- "How to Customize Browse Page Tables" on the next page

How to Customize Browse Page Search Criteria

Browse pages contain a search component enabling you to quickly find the artifacts in the table of artifacts. There are multiple search components to correspond with each filter in a browse page.

To demonstrate the functionality of these elements the following example describes the creation of search criteria for the All filter for SLOs:

To create search criteria:

1. Edit the **SLOs** browse filters, as described in "How to Customize Browse Page Filters" on page 121.

2. Click the **All** filter to refresh the page with the search criteria for that filter.

3. Click **Customize** inside the search box to open the XML representation of the search criteria.

4. Replace the default content with the following XML extract:

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
      xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="cust artifactView.xsd">
  <search>
    <criteria id="nameAndDescription" label="" type="properties" visible="true"
          widget="search:nameAndDescription">
      <parameter name="name">name</parameter>
      <parameter name="name">description</parameter>
    </criteria>
    <criteria id="timing" label="Timing Required" type="property"
visible="false">
      <parameter name="name">s.timingRequired</parameter>
    </criteria>
  </search>
</customization>
```

   For more details, see "Search Criteria Customization Reference" on page 161.

5. Click **Save** to create the SLOs search criteria for the All filter.

6. Repeat this procedure selecting the **Favorites** filter with an appropriate criteria customization.

In this example, the main text box query is for name and description and the Timing Required property is now available as an additional criteria for the search. Set `visible=true` to make additional criteria appear by default in the search component.

Notice that the `timing` criteria uses a property of SLOs defined by the SDM model and the alias set in the filter customization. For more details about DQL queries, see "Using DQL" in the *Developer Guide*.

How to Customize Browse Page Tables

Browse pages contain a table listing artifacts primarily determined by the datasource query from the selected filter and then any specified search criteria from the search component.

The table consists of the following elements:

- A set of facets, enabling you to reduce the table content according to the selectable criteria.
- A table of artifacts.

To demonstrate the functionality of these elements the following example describes the creation of a results table for the All filter for the Browse SLOs page:

To create a table:

1. Edit the **SLOs** browse filters, as described in "How to Customize Browse Page Filters" on page 121.
2. Click the **All** filter to refresh the page with the table for that filter.
3. Click **Customize** in the table component to open the XML representation of the table.
4. Replace the default content with the following XML extract:

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
      xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="cust table.xsd">
  <table>
    <column filterBy="s.name" id="name" label="Name" sortBy="s.name">
      <content componentName="/core/util/artifactLinkRenderer">
        <parameter name="uuid" queryColumn="s._uuid"/>
        <parameter name="caption" queryColumn="s.name"/>
        <parameter name="sdmName" queryColumn="s._sdmName"/>
      </content>
    </column>
    <rowPreview id="description">
      <content queryColumn="s.description"/>
    </rowPreview>
  </table>
  <facets>
    <facet id="impactFacet" label="Business Impact"
queryColumn="s.businessImpact" type="category"/>
  </facets>
</customization>
```

   For more details, see "Table Customization Reference" on page 149.

5. Click **Save** to create your SLOs table.
6. Repeat this procedure selecting the **Favorites** filter with an appropriate table customization.

In this example, the table contains a link to each SLO artifact using the core `artifactLinkRenderer` component and the description property and a facet for filtering the table by Business Impact categories.

Notice that the table content is defined by SDM property names and the alias set in the filter query. For more details about DQL queries, see "Using DQL" in the *Developer Guide*.

How to Customize Create / Edit Artifact Pages

Clicking **Create** in the **Artifact** menu Create Artifact page for the artifact type. Access Edit Artifact pages from the context actions component in on the right side of artifact detail pages.

The customization for a create/edit artifact page is very similar to the customization for the artifact detail pages. There are two customization links at the top-left, the first contains the page layout which usually only contains the `/detail/editArtifactProperties` component and should not need any customization. The second link refers to the artifact content and is similar to the customization for the artifact content in artifact detail pages. The customization usually consists of a set of artifact properties and related artifact tables with add/upload functionality.

To customize a create/edit artifact page:

1. In the customization view of the role specific UI, navigate to the new/edit artifact page you want to modify, and click the second **Customize** at the top-left of the page.

   An XML editor window opens displaying the XML extract for the page content.

   The customization contains a set of property and component elements that are usually organized into groups with label headings.

2. Modify the XML extract as required.

   For details, see "Artifact Content Customization Reference" on page 146

3. Click **Save** to make your changes.

The following example extends the existing Create SLO page with the addition of the Business Impact property and a Documentation table with Add, Upload, and Delete functionality.

To add a Documentation table to the Create SLO page:

1. Click the **SLO** link in the **Create** section of the menu as described in “"How to Customize Menus" on page 114.

2. Click **SLO** to the existing Create SLO page.

3. Click the second **Customize** link to open the default XML representation of the content of the page.

4. Add the following property element between description and keywords to add the Business Impact taxonomic property:

   ```
   <property name="businessImpact" id="businessImpact"/>
   ```

5. Add the following XML extract after the Objectives group to add a Documentation table at the bottom of the page:

   ```
   <group id="documentation" label="Documentation">
     <component componentName="/core/relations/genericRelationshipComponent"
   id="documentation.table">
       <parameter name="tableDefinitionId">${parentCustomizationId}.documentation
                          </parameter>
       <parameter name="relation">documentation</parameter>
       <parameter name="artifact">${artifact}</parameter>
     </component>
   </group>
   ```

   The most significant element in this extract is `<parameter name="relation">`. Use the SDM name of the relationship in this element to define which set of related artifacts the table handles.

6. Click **Save** to add property and the table to the page content.

   This adds the property and the default documentation table which does not contain the required functionality and requires further customization.

7. Click **Customize** in the Documentation table to open the XML representation of the table.

8. Replace the content of the table customization with the following XML extract:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
         xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="cust table.xsd">
     <table selectionModel="multiple" visible="ifNotEmpty">
       <rowId queryColumn="_uuid"/>
       <column filterable="false" id="fav" label="" sortable="false">
         <content componentName="/core/favorites/addFavorite">
           <parameter name="uuid" queryColumn="_uuid"/>
           <parameter name="mode">table</parameter>
         </content>
         <property name="menuDisabled">true</property>
         <property name="fixed">true</property>
         <property name="width">30</property>
       </column>
       <column filterBy="name" id="name" label="Name" sortBy="name">
         <content componentName="/core/util/artifactLinkRenderer">
           <parameter name="uuid" queryColumn="_uuid"/>
           <parameter name="caption" queryColumn="name"/>
           <parameter name="mode">data</parameter>
         </content>
   ```

```
    </column>
    <column id="type" label="Document Type">
      <content queryColumn="documentType.name"/>
       </column>
    <column id="version" label="Version">
      <content queryColumn="version"/>
      <property name="hidden">true</property>
    </column>
    <column id="date" label="Date">
      <content queryColumn="_revisionTimestamp"/>
       </column>
       <rowPreview id="description">
      <content queryColumn="description"/>
    </rowPreview>
  </table>
  <actions>
    <actionComponent action="addDocumentArtifact"
          componentName="/core/documentation/uploadDocument"
          description="Upload a local file to the repository"
id="addDocumentArtifact"
          label="Upload" styleClass="Upload">
      <parameter name="artifactType">DocumentationArtifact</parameter>
      <parameter name="artifact">${artifact}</parameter>
      <parameter name="datasource">${datasource}</parameter>
    </actionComponent>
    <actionComponent action="addRelation"
componentName="/core/relations/addRelation"
          description="Add an existing artifact" id="addRelation" label="Add"
styleClass="Add">
      <parameter name="relation">${relation}</parameter>
      <parameter name="artifact">${artifact}</parameter>
      <parameter name="datasource">${datasource}</parameter>
    </actionComponent>
    <actionComponent action="removeRelation"
componentName="/core/relations/removeRelation"
          description="Delete selected items" id="removeRelation"
label="Delete" styleClass="Remove">
      <parameter name="selectionBag">${selectionBag}</parameter>
      <parameter name="relation">${relation}</parameter>
      <parameter name="artifact">${artifact}</parameter>
      <parameter name="datasource">${datasource}</parameter>
    </actionComponent>
  </actions>
</customization>
```

This XML is a copy of the customization for the Documentation table in the Create Service page and requires no modification to function in the Create SLO page.

9. Click **Save**. The table now contains Add, Delete, and Upload functionality.

How to Customize Artifact Detail Pages

Clicking an artifact name opens a detail page for the artifact. Depending on the artifact type, the page contains a set of artifact properties, related artifact tables, context actions, a number of components and thirteen tabs offering different views of the artifact.

> **Caution:** There must be an instance of the artifact type accessible to users in the UI in order to be able to navigate to the artifact details page.

The default artifact detail page consists of the following customizable elements:

- Click **Customize** at the top-left of the page to modify the overall layout of the page.

  For details, see "How to Customize Artifact Detail Tab Layouts" below.

- Click **Customize** above the artifact name to customize the set of artifact properties and related artifacts shown for the artifact type.

  For details, see "How to Customize Artifact Detail Page Properties" on the next page.

- Click **Customize** in a related artifact table to customize the table for that artifact type.

  For details, see "How to Customize Related Artifact Tables" on page 132.

- Click **Customize** in a component to customize that component, for example, context actions, projects, and contacts.

  For example, see "Context Actions Customization Reference" on page 157.

How to Customize Artifact Detail Tab Layouts

Artifact detail pages consist of two columns, containing artifact properties, related artifact tables, and a set of components.

To customize a detail tab layout:

1. In the customization view of the role specific UI, navigate to the tab page you want to modify, and click **Customize** in the top left of the page under the tab name.

   An XML editor window opens displaying the XML extract for the page.

2. Modify the XML extract as required.

   The XML extract for a page layout may contains the following customizable elements:

- An artifact details component in the left column.
- A set of components in the right column.
- A table

3. Click **Save** to save your changes locally.

How to Customize Artifact Detail Page Properties

The properties and related artifacts shown for an artifact are determined by the artifact details component.

To customize a detail page component:

1. In the customization view of the role specific UI, navigate to the detail page you want to modify, and click **Customize** for the component you want to modify.

   An XML editor window opens displaying the XML extract for the component.

   For example, in the detail page for an SLO, click **Customize** above the SLO name to modify its details layout.

2. Modify the XML extract as required.

   For details, see "Artifact Content Customization Reference" on page 146.

   For example, add the following extract to the `general` element of the SLO details XML representation:

   ```
   <property name="businessImpact" id="businessImpact"/>
   ```

   Adding a table of related artifacts is the same as described in the example in “Customizing Create/Edit Artifact Pages”.

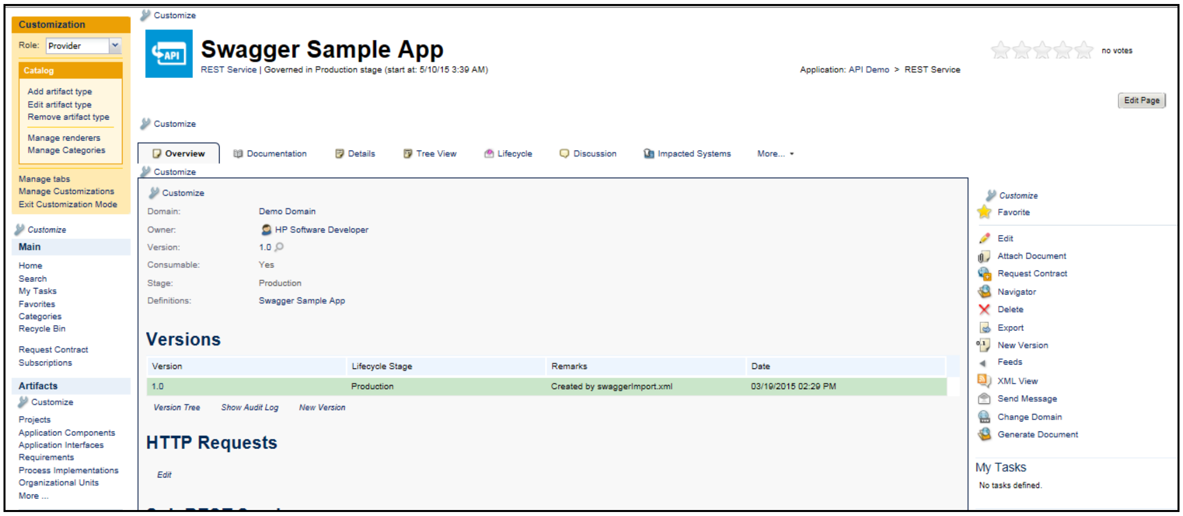
3. Click **Save** to make your changes.

   In this example, Business Impact and the value for the SLO appears in the SLO properties section of the page.

How to Customize Artifact Detail Pages Breadcrumbs

Artifact instances could be associated with multiple instances of other artifact types, for example, Rest Service with multiple HTTP Requests, HTTP Endpoints, Swaggers and WADLs. When user is in the View page of a certain artifact instance, he can navigate back to its parent artifacts through **Breadcrumb** path.

By default, breadcrumb path of artifact instance will be displayed with full artifact type and artifact name of its parents. But this displaying is customizable.



Swagger Sample App

REST Service | Governed in Production stage (start at: 5/10/15 3:39 AM)

Application: API Demo > REST Service

Domain: Demo Domain
Owner: HP Software Developer
Version: 1.0
Consumable: Yes
Stage: Production
Definitions: Swagger Sample App

Versions

Version	Lifecycle Stage	Remarks	Date
1.0	Production	Created by swaggerImport.xml	03/19/2015 02:29 PM

HTTP Requests

To customize the displaying of breadcrumb path:

1. Open the **Artifact Detail** page of an artifact that has the breadcrumb path.

2. Click **Customize** link in the header.

3. To change the displaying type of breadcrumb, modify value of **showLable** parameter:

   a. **both**: display both artifact type and artifact name in breadcrumb path

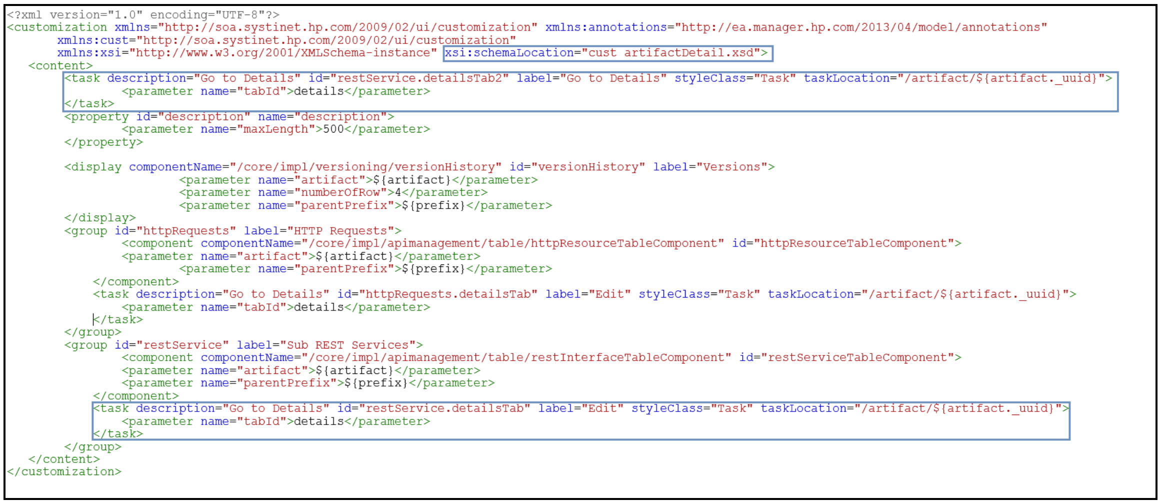
   b. **name**: display only artifact name in breadcrumb path

   c. **type**: display only artifact type in breadcrumb path

```
<?xml version="1.0" encoding="UTF-8"?>

<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="cust
header.xsd">

<breadcrumb>

<parameter name="showLabel">both</parameter>

</breadcrumb>

</customization>
```

How to Customize Artifact Detail using Task tags

Using Task tag in customization of Artifact Details page:



```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization" xmlns:annotations="http://ea.manager.hp.com/2013/04/model/annotations"
       xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="cust artifactDetail.xsd">
   <content>
        <task description="Go to Details" id="restService.detailsTab2" label="Go to Details" styleClass="Task" taskLocation="/artifact/${artifact._uuid}">
                <parameter name="tabId">details</parameter>
        </task>
        <property id="description" name="description">
                <parameter name="maxLength">500</parameter>
        </property>

        <display componentName="/core/impl/versioning/versionHistory" id="versionHistory" label="Versions">
                        <parameter name="artifact">${artifact}</parameter>
                        <parameter name="numberOfRow">4</parameter>
                        <parameter name="parentPrefix">${prefix}</parameter>
        </display>
        <group id="httpRequests" label="HTTP Requests">
                <component componentName="/core/impl/apimanagement/table/httpResourceTableComponent" id="httpResourceTableComponent">
                <parameter name="artifact">${artifact}</parameter>
                        <parameter name="parentPrefix">${prefix}</parameter>
            </component>
            <task description="Go to Details" id="httpRequests.detailsTab" label="Edit" styleClass="Task" taskLocation="/artifact/${artifact._uuid}">
                <parameter name="tabId">details</parameter>
            </task>
        </group>
        <group id="restService" label="Sub REST Services">
                <component componentName="/core/impl/apimanagement/table/restInterfaceTableComponent" id="restServiceTableComponent">
                <parameter name="artifact">${artifact}</parameter>
                <parameter name="parentPrefix">${prefix}</parameter>
            </component>
            <task description="Go to Details" id="restService.detailsTab" label="Edit" styleClass="Task" taskLocation="/artifact/${artifact._uuid}">
                <parameter name="tabId">details</parameter>
            </task>
        </group>
   </content>
</customization>
```

The task tag should be added in <**content**> or <**group**> tag.

Attributes of task tag are:

- `label`: Label will be shown on the UI.
- `description`: A description includes rich text and HTML support.
- `styleClass`: It is the name of css class.
- `taskLocation`: It is path of expected page to be navigated. (required)
- `id`: It is the unique id (required)

User can add “parameter” tag in “task” label and “parameter” tags will be used to query string parameters in URL.

Examples:

Navigate to Details tab:

```
<task description="Go to Details" id="restService.detailsTab2" label="Go to
   Details"

styleClass="Task" taskLocation="/artifact/${artifact._uuid}">

   <parameter name="tabId">details</parameter>

</task>
```

Navigate to Reports tab:

```
<task id="reports.tab" label="Reports" taskLocation="/reports/homepage"/>
```

How to Customize Related Artifact Tables

Many artifact instances are associated with multiple instances of other artifact types, for example, Services with multiple Implementations, Documentation, and SLOs. These relationships appear in Artifact Detail pages as tables of related artifacts. Each table of artifacts is customizable.

To customize a table of related artifacts:

1. Open the artifact detail page containing the table you want to customize.

2. Click **Customize** for the table you want to edit to open the XML representation of the table.

   Tables can consist of the following elements:

   - A datasource defining the content of the table. In artifact detail page related artifact tables there is usually no datasource as the content is defined in the artifact detail page properties customization by the relationship between the artifact in the page and the artifact type in the table.

     For details, see "Datasource Customization Reference" on page 150

   - A table defining the behaviour and content of the table. Set functionality such as sorting and selection in this element.

     For details, see "Table Layout Customization Reference" on page 151

   - An optional set of actions defining functionality for the artifact type in the table such as add and delete.

     For details, see "Table Actions Customization Reference" on page 155

   - An optional set of facets that act as selectable filters for the table content.

     For details, see "Table Facets Customization Reference" on page 156

3. Modify the XML as required.

4. Click **Save** to save your changes locally.

The following example adds the Business Impact taxonomic property to the SLOs table in the Service artifact detail page as a sortable table column and as a selectable facet for the table.

To add Business Impact to the SLO table:

1. Open the artifact detail page for a service.

2. Click **Customize** in the SLOs table to open its XML representation.

3. To add Business Impact as a sortable column in the table, add the following XML extract before

the <rowPreview> element of the table:

```
<column id="impact" label="Impact">
  <content queryColumn="a.businessImpact.name"/>
</column>
```

Note the use of `.name` as Business Impact is a taxonomic property.

4. To add Business Impact as a selectable facet to filter the table, add the following XML extract before the closing </customization> tag:

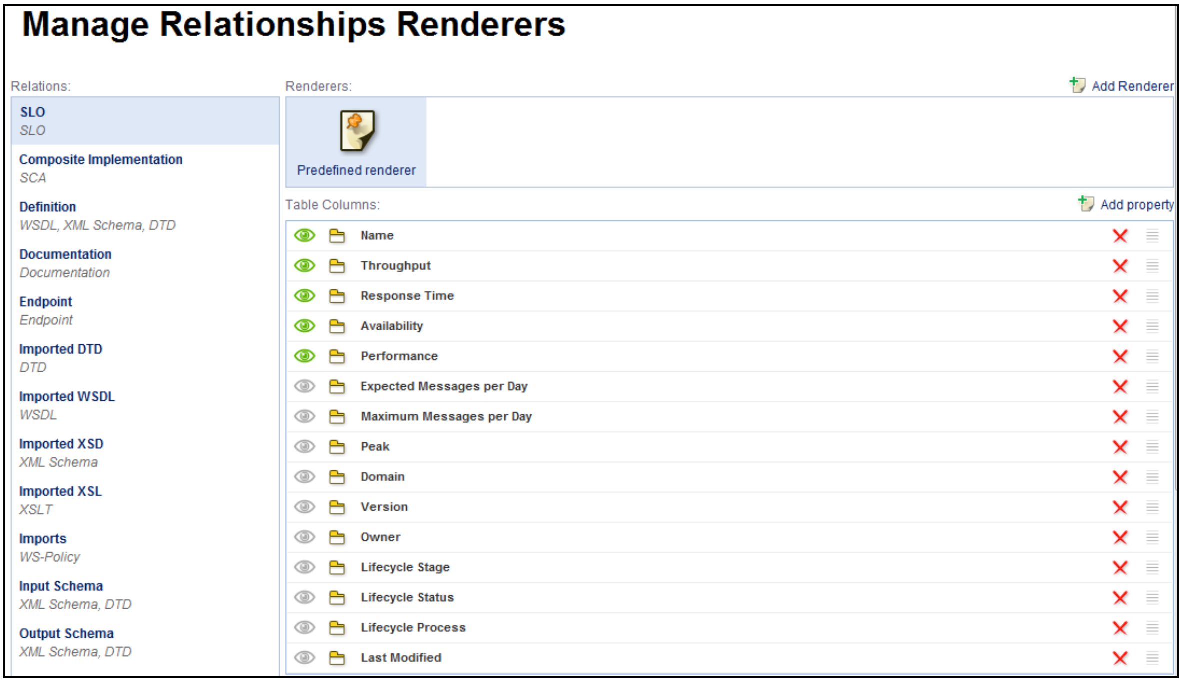
```
<facets>
  <facet id="impactFacet" label="Business Impact"
queryColumn="a.businessImpact" type="category">
    <parameter name="propertyName">a.businessImpact</parameter>
  </facet>
</facets>
```

How to Customize Renderers

Relationship renderers provide views of relationships between artifacts. All relations are provided with a default predefined renderer that can be modified or a new renderer can be defined.

Click **Manage Renderers** from the Customization menu to open the Renderer Management page.

Screenshot: Managing Renderers



Manage Relationships Renderers

Relations:

SLO
SLO

Composite Implementation
SCA

Definition
WSDL, XML Schema, DTD

Documentation
Documentation

Endpoint
Endpoint

Imported DTD
DTD

Imported WSDL
WSDL

Imported XSD
XML Schema

Imported XSL
XSLT

Imports
WS-Policy

Input Schema
XML Schema, DTD

Output Schema
XML Schema, DTD

Renderers:

Add Renderer

Predefined renderer

Table Columns:

Add property

Name
Throughput
Response Time
Availability
Performance
Expected Messages per Day
Maximum Messages per Day
Peak
Domain
Version
Owner
Lifecycle Stage
Lifecycle Status
Lifecycle Process
Last Modified

Clicking on a relation will present the predefined renderer for that relation.

Modifications available to a renderer are:

- Making the existing properties visible or invisible by clicking on the eye icon.
- Adding a property by clicking on the **Add property** link and selecting one or more properties from the popup list.
- Dragging and dropping the properties to change the order.
- Renaming the property by clicking on the property name.
- Deleting the property by clicking the red X icon.

To add a new renderer, click on the **Add Renderer** link, give the renderer a name and click **Add**.

The new renderer can be modified in exactly the same manner as the predefined renderer or deleted by clicking the **Delete Renderer** link.

Note: Predefined renderers cannot be deleted.

How to Customize Dependency Analysis table

To customize a dependency analysis table layout:

1. In the customization view of the role specific UI, navigate to the tab page you want to modify, and click **Customize** in the top left of the page under the tab name.

   An XML editor window opens displaying the XML extract for the page.

2. Modify the XML extract as required.
3. Click **Save** to save your changes locally.

How to Customize Impacted Systems table

To customize a impacted systems table layout:

1. In the customization view of the role specific UI, navigate to the tab page you want to modify, and click **Customize** in the top left of the page under the tab name.

   An XML editor window opens displaying the XML extract for the page.

2. Modify the XML extract as required.
3. Click **Save** to save your changes locally.

How to Customize Categories

Within each role-based UI, categories are available for browsing or selection in the following places:

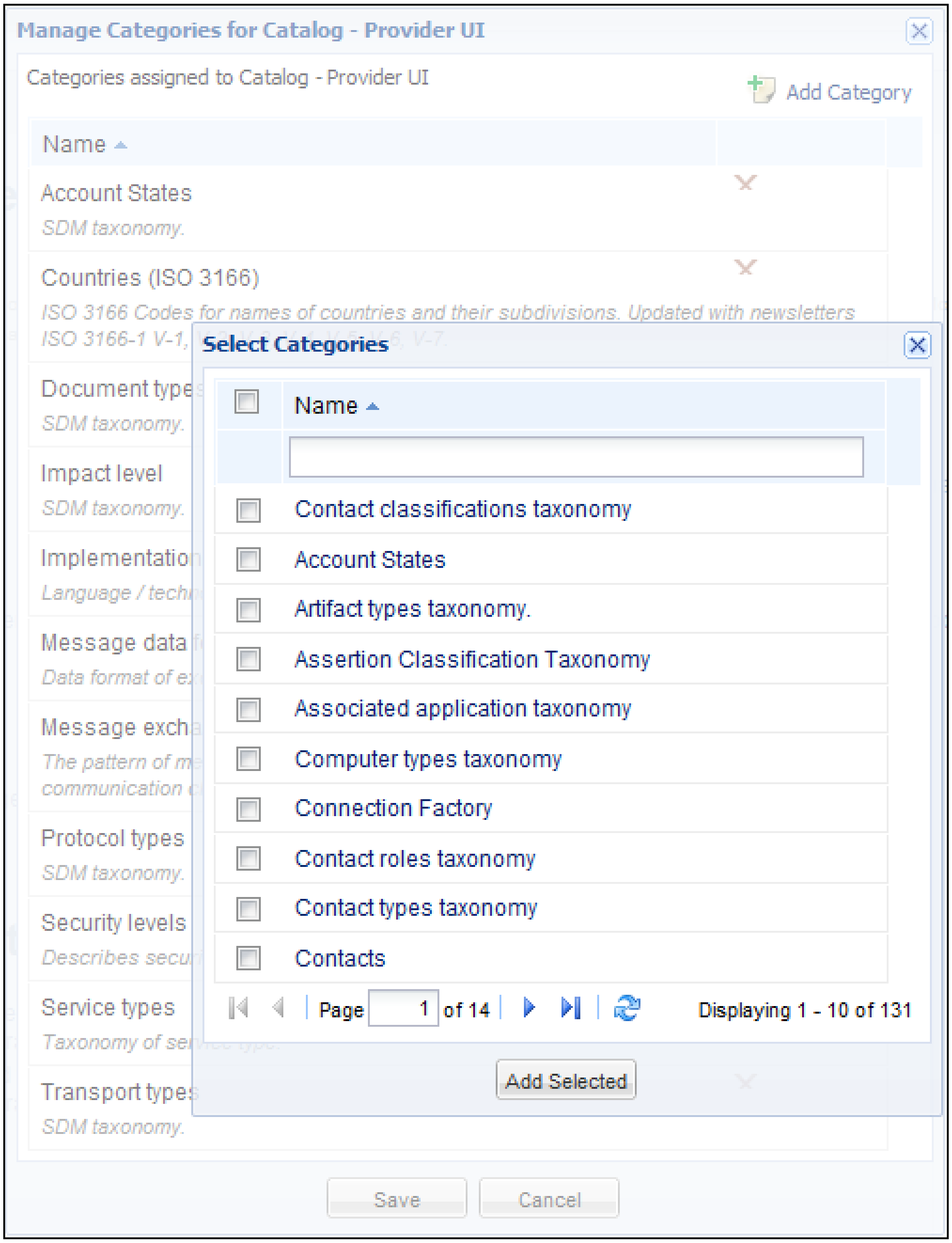
- The Categories component on the homepage.
- The Categories page accessible from the **View All Categories** link.
- The Add Categories function available in Create Artifact pages and as criteria in Search pages and components.

Each of these categories displayed is a predefined set of taxonomies which you can customize in the standard UI.

To customize categories:

1. In the Administration tab, click **Customize** to open the customization menu.
2. Click **Manage Categories** to open the dialog for the currently selected role tab
3. Clicking **Add Category** will present a list of categories that be added and once selected, click **Add Selected** to make them visible. Categories can be deleted by clicking the red **X** on the right hand side of the named row.

Screenshot: Manage Categories



Manage Categories for Catalog - Provider UI

Categories assigned to Catalog - Provider UI

Name ▲

Account States
SDM taxonomy.

Countries (ISO 3166)
ISO 3166 Codes for names of countries and their subdivisions. Updated with newsletters ISO 3166-1 V-1, V-2, V-3, V-4, V-5, V-6, V-7.

Document type
SDM taxonomy.

Impact level
SDM taxonomy.

Implementatio
Language / techn

Message data f
Data format of e

Message excha
The pattern of me communication c

Protocol types
SDM taxonomy.

Security levels
Describes secur

Service types
Taxonomy of service type.

Transport types
SDM taxonomy.

Select Categories

□ Name ▲

□ Contact classifications taxonomy
□ Account States
□ Artifact types taxonomy.
□ Assertion Classification Taxonomy
□ Associated application taxonomy
□ Computer types taxonomy
□ Connection Factory
□ Contact roles taxonomy
□ Contact types taxonomy
□ Contacts

Page 1 of 14

Displaying 1 - 10 of 131

4. Click **Save**.

How to Manage UI Customizations

The Manage Customizations page lists all role-based UI customizations, both Public and Private. Access the page by clicking the **Manage Customizations** link in the Customization menu in customization mode.

The Manage Customizations page contains a table of unreleased customizations,an import tool, a delete option and a history.

The table of customizations contains the following functionality:

- **Release**

  Select private customizations, and click **Release** to make them public. All users in the appropriate role view these customizations the next time they log in.

- **Delete**

  Select private customizations, and click **Delete** to remove them from your local version.

- **Restore**

  Select public customizations, and click **Restore** to remove these customizations and change to the default version.

Use the import functionality to transfer customizations between different Systinet servers.

To import UI customizations:

1. In the Manage Customizations page, click **Import** and then **Browse** to locate your customization archive, or input the full path.

2. Click **Import** to upload the selected customization archive.

Tab Customizations enable you to modify the configuration of the tabs for particular user roles, or create new tabs for existing perspectives.

For details, see "Tab Configuration Reference" on page 184 and "How to Create Role-Based Tabs" below

How to Create Role-Based Tabs

As well as customizing the default tabs provided for the Business Analyst and Business Partner, you can create additional tabs for specified user perspectives.

To create a role-based tab:

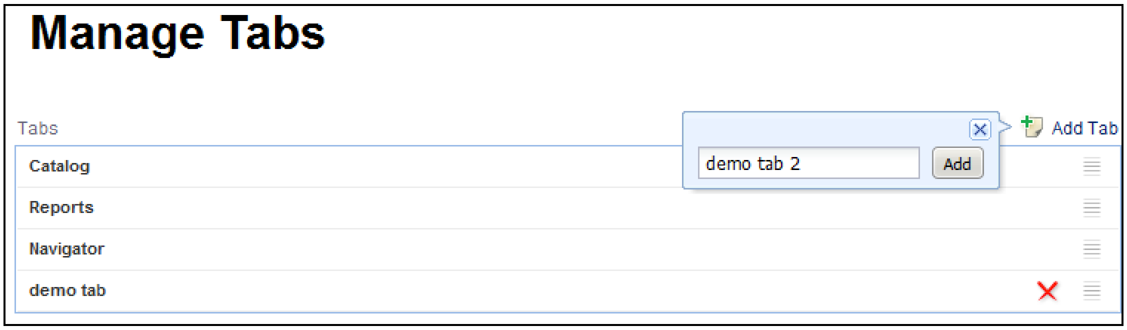
1. From the Administrator Tab, start Customization mode by clicking the **Customize** link.

2. Select the role from the drop-down list for which a new tab is required and click **Manage tabs**.

   A list of existing tabs for this role is displayed.

3. Click the **Add Tab** button, enter a name for the tab and click **Add** to create the tab.

Screenshot: Managing Tabs



Manage Tabs

The location of the tab can be moved by clicking and holding on the far right of the named row and dragging it to the desired location. Upon refreshing the page, the tab will be visible in the selected location.

Clicking the red X at the far left of the named will delete the tab.

> **Note:** Default tabs cannot be deleted.

4. Create the pages for your new tab. For ease of customization, HP recommends copying equivalent customizations from another tab of the same type, pasting them into the new tab and making your required changes.

   HP suggests making your customizations in the following order:

   a. Customize the menu to create a set of navigation links enabling you to open the pages you require.

      For details, see "How to Customize Menus" on page 114 and "Menu Customization Reference" on page 142.

   b. Customize the homepage.

      For details, see "How to Customize Homepages" on page 116 and "Page Layout Customization Reference" on page 144.

   c. Use the View links in your menu to access the Browse pages you want to customize.

      For details, see "How to Customize Browse Pages" on page 118 and "Page Layout Customization Reference" on page 144.

   d. Use the New links in your menu to access the Create pages you want to customize.

      For details, see "How to Customize Create / Edit Artifact Pages" on page 125 and "Page Layout Customization Reference" on page 144.

   e. To customize artifact detail pages you need to create an artifact of that type. Either import a data set with artifacts visible to the perspective, or create a set of New links in the menu so that you can create artifacts.

Click the View link to open the browse page for the artifact type you want to modify, and then click an artifact name to open the artifact detail page.

For details, see "How to Customize Artifact Detail Pages" on page 128 and "Page Layout Customization Reference" on page 144.

f. Use the Edit context action in your artifact detail pages to access the Edit pages you want to customize.

For details, see "How to Customize Create / Edit Artifact Pages" on page 125 and "Page Layout Customization Reference" on page 144.

UI Customization Reference

This section contains a reference guide to customizing role-based UIs in the following sections:

- "Generic XML Customization" below
- "Customizable Components Reference" on the next page
- "Reusable UI Components Reference" on page 170
- "Reusable UI Tasks Reference" on page 183
- "Tab Configuration Reference" on page 184

Generic XML Customization

A typical UI customization XML contains the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://systinet.hp.com/2009/02/ui/customization"
      xmlns:cust="http://systinet.hp.com/2009/02/ui/customization"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="cust table.xsd">
  ...
</customization>
```

- An XML definition, with recommended UTF-8 encoding.
- A root customization element.

  This element must use namespace `http://systinet.hp.com/2009/02/ui/customization` and must have an associated XML schema, in this example, `table.xsd`.

In many places in customization XMLs, you can use JSTL expression language to provide runtime values. For example, in an artifact detail page you can use expression `${artifact._uuid}` as a datasource parameter. Systinet replaces this parameter during runtime with the UUID of the current artifact.

If you use another customizable component in a customization, you must provide a `customizationId` parameter for it. You can use the following type of expression to provide a parameter:

```
<parameter name="customizationId">${parentCustomizationId}.aboutText</parameter>
```

This expression means, use the customization ID of the component and add `.aboutText` to it. If you use this in the analyst homepage (with customization ID `analyst.homepage` the included component has customization ID parameter `analyst.hompage.aboutText`.

Customizable Components Reference

Each page in a role-based UI consists of a set of customizable elements which may contain further customizable elements.

this section describes the main customizable elements, their typical XML, and the attributes for each XML element, in the following sections:

- "Menu Customization Reference" on the facing page
- "Page Layout Customization Reference" on page 144
- "Artifact Content Customization Reference" on page 146
- "Table Customization Reference" on page 149
- "Context Actions Customization Reference" on page 157
- "Query Selector Customization Reference" on page 159
- "Search Criteria Customization Reference" on page 161
- "Search Component Customization Reference" on page 164
- "HTML Areas Customization Reference" on page 168

Menu Customization Reference

Role-based UIs display a navigation menu on the left of all pages. The menu consists of grouped items and components and must conform to the `navigation.xsd` schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
    xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="cust navigation.xsd">
  <navigation>
    <group id="analyst.view" label="View">
      <task id="analyst.view.home" label="Home"
          taskLocation="/analyst/homepage"/>
      <task id="analyst.view.favorites" label="Favorites"
          taskLocation="/common/favorites"/>
      <ruler id="analyst.view.ruler1"/>
      <task id="analyst.view.services" label="Services"
          taskLocation="/analyst/browseArtifact">
        <parameter name="artifactType">businessServiceArtifact</parameter>
      </task>
      ...
    </group>
    <group id="analyst.new" label="New">
      <task id="analyst.new.service" label="Service"
          taskLocation="/analyst/newArtifact">
        <parameter name="sdmName">businessServiceArtifact</parameter>
      </task>
      ...
    </group>
    <group id="analyst.recent" label="Recent Documents">
      <component componentName="/core/recentlyUsedArtifacts"
          id="analyst.recent.list"/>
    </group>
  </navigation>
```

```
</customization>
```

The customization XML for the menu contains the following tags with their attributes and parameters:

XML Element	Description • Contains elements	Attributes	Description
<navigation>	Containing element for the menu. <group>		
<group>	Provides a visual grouping of navigation items. • <task> • <component> • <ruler>	id	A unique group identifier.
		label	The group display name.
<task>	Defines a link to a framework task defined in the framework configuration file (`webui.xml`). <parameter>	id	Unique task identifier.
		label	The link label.
		taskLocation	The target task URI.
<component>	Includes a more complex piece of functionality into the menu (for example, Recent Documents). <parameter>	componentName	The component identification as defined in the web framework configuration file (`webui.xml`).
		id	A unique component placement identifier.
<ruler>	Display an item separator line.	id	A unique ruler identifier
<parameter>	Optional name and value parameter pairs to pass to the target task or component.	name	Parameter name with the value expressed as the parameter element content.

Page Layout Customization Reference

Page layouts define the content of pages. They are typically a set of components organized into a one or two column layout. Page customizations must conform to `columns.xsd`.

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="cust columns.xsd">
  <columns>
    <column id="leftColumn">
      <component componentName="/core/text/editableRichTextArea" id="leftColumn.about">
        <parameter name="customizationId">${parentCustomizationId}.aboutText</parameter>
        <parameter name="title">About Catalog</parameter>
      </component>
      <component componentName="/core/search/homepageSearch" id="leftColumn.search">
        <parameter name="customizationId">${parentCustomizationId}.suggestedSearch
               </parameter>
        <parameter name="viewId">analyst</parameter>
      </component>
      ...
    </column>
    <column id="rightColumn">
      <component componentName="/core/contract/myContracts" id="rightColumn.mycontracts">
        <parameter name="numberOfContracts">5</parameter>
      </component>
      <component componentName="/core/favorites/favoriteList" id="rightColumn.favorites"/>
      ...
    </column>
  </columns>
</customization>
```

The customization XML for pages contain the following tags with their attributes and parameters:

XML Element	Description • Contains elements	Attributes	Description
<columns>	Containing element for pages. • <column>		
<column>	Defines a page column with a unique ID. There must be one or two columns in a page. • <component>	id	A unique group identifier.
<component>	Includes a component into the column. Components appear in the same order as they are listed in the customization. • <parameter>	competent	The component identification as defined in the web framework configuration file (`webui.xml`). You can have multiple occurrences of the same component on a page but they must have different IDs.
		id	A unique group identifier.
<parameter>	Optional name and value parameter pairs to pass to the target component.	name	Parameter name with the value expressed as the parameter element content.

Role Customization Reference

The example syntax is:

```
<contextActions>
  <task id="navigator" label="Navigator" taskLocation="/common/navigator">
    <parameter name="uuid">${artifact._uuid}</parameter>\
    <restriction allowOwner="true">
      <visibleFor>ROLE1</visibleFor>
      <visibleFor>ROLE2</visibleFor>
    </restriction>
  </task>
```

```
</contextActions>
```

This declaration causes the "Navigator" link to be only displayed to users wth ROLE1 or ROLE2. Note, that some built-in roles depend on licenses present (for example, PolicyDeveloper or ContractRequestor).

Items that can be restricted (inner element restriction in the customization XML):

- parameter
- property
- task
- group
- component

Visible for nested elements list required role(s), allowOwner flag allows access if the user owns the current artifact (e.g. when the artifact detail is being viewed).

Artifact Content Customization Reference

Artifact detail pages are typically organized into two columns with components in the right column and an the `/detail/viewArtifactProperties` component in the left column. You can access and customize the artifact content displayed in the left column with the **Customize** link at the top of the column. New Artifact pages usually contain the similar content to the left column of detail pages. The customization typically consists of components and properties organized into groups.

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="cust artifactDetail.xsd">
  <content>
    <property id="description" name="description"/>
    <group id="properties" label="">
      <property id="version" name="version"/>
      <property id="criticality" name="criticality"/>
      <component componentName="/core/contact/providerWrapper" id="provider">
        <parameter name="artifact">${artifact}</parameter>
        <parameter name="customizationId">${viewId}.contacts.suggestedSearch</parameter>
      </component>
      <display componentName="/core/util/parentLinksRenderer" id="parentArtifacts" label="Parent Artifacts">
        <parameter name="uuid">${artifact._uuid}</parameter>
        <parameter name="sdmName">${artifact._artifactSdmName}</parameter>
      </display>
    </group>
    <group id="documentation" label="Documentation">
      <component componentName="/core/relations/genericRelationshipComponent" id="documentation.table">
        <parameter name="tableDefinitionId">${parentCustomizationId}.documentation
               </parameter>
        <parameter name="relation">documentation</parameter>
        <parameter name="artifact">${artifact}</parameter>
      </component>
    </group>
  </content>
  </customization>
```

The customization XML for artifact content contains the following tags with their attributes and parameters:

XML Element	Description Contains elements	Attributes	Description
<content>	Containing element for artifact content. • <group> • <property> • <component> • <display>		

XML Element	Description Contains elements	Attributes	Description
<group>	Surrounds a set of properties or components in a labelled colored block. • <property> • <component> • <display>	id	A unique group identifier.
		label	An optional label for the group.
<property>	Displays and artifact property. There must be a defined widget for the property in the widgets configuration.	id	A unique property display ID.
		name	The SDM name of the property.
		label	An optional property label. If a label is not specified the label from the SDM is used.
<component>	Includes a more complex piece of functionality, for example a table of related artifacts. <parameter>	componentName	The component identification as defined in the web framework configuration file (`webui.xml`). You can have multiple occurrences of the same component on a page but they must have different IDs.
		id	A unique component placement identifier.
<display>	Provides the same functionality as a component with the additional option to add a label before the component. <parameter>	componentName	The component identification as defined in the web framework configuration file (`webui.xml`). You can have multiple occurrences of the same component on a page but they must have different IDs.
		id	A unique component placement identifier.
		label	An optional label for the component.

XML Element	Description Contains elements	Attributes	Description
<parameter>	Optional name and value parameter pairs to pass to the target component.	name	Parameter name with the value expressed as the parameter element content.

Table Customization Reference

Systinet uses table components in many pages in the role-based UI. These tables display the results of a query provided by a datasource and provide filtering, sorting, and paging functionality. Table customizations must conform to `table.xsd`.

Table customizations contain a number of elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="cust table.xsd">
  <datasource>
    ...
  </datasource>
  <table>
    ...
  </table>
  <facets>
    ...
  </facets>
  <actions>
    ...
  </actions>
</customization>
```

The following sections describe each of these elements:

- "Datasource Customization Reference" below
- "Table Layout Customization Reference" on the next page
- "Table Actions Customization Reference" on page 155
- "Table Facets Customization Reference" on page 156

Datasource Customization Reference

A datasource provides a set of query results for a table. You can put the datasource in a table customization or alternatively, put it in another component customization (for example, search criteria) and pass it as a parameter to the table.

```
<datasource>
   <type>dql</type>
   <query>select from businessServiceArtifact where b._uuid = :serviceUuid</query>
   <parameter name="serviceUuid">${artifact._uuid}</parameter>
</datasource>
```

The customization XML for a datasource contains the following tags with their attributes and parameters:

XML Element	**Description** • **Contains elements**	Attributes	Description
<datasource>	Containing element for a datasource. • <type> • <query> • <parameter>		
<type>	An optional element which defaults to `dql`.		

XML Element	Description • Contains elements	Attributes	Description
<query>	A shortcut for <parameter name="query"> containing the actual query.		
<parameter>	Optional name and value parameter pairs to replace query content specified by `:parametername`.	name	Parameter name with the value expressed as the parameter element content.

For more details about DQL queries, see "Using DQL" in the *Developer Guide*.

> **Note:** Table components do not maintain the datasource instance. It is created on each request from the configuration and must be maintained by the calling component if it needs to be maintained in a session.

Table Layout Customization Reference

The <table> element of the customization defines the layout and functionality of the table itself.

```
<table visible="ifNotEmpty" selectionModel="multiple" filter="auto" pageSize="10">
  <rowId queryColumn="i._uuid"/>
  <column id="name" label="Name" sortBy="a.name" filterBy="a.name">
    <content componentName="/core/util/artifactLinkRenderer">
      <parameter name="uuid" queryColumn="a._uuid"/>
      <parameter name="caption" queryColumn="a.name"/>
    </content>
  </column>
  <column id="version" label="Version">
    <content queryColumn="a.version"/>
    <property name="hidden">true</property>
  </column>
  <rowPreview id="description">
    <content queryColumn="a.description"/>
  </rowPreview>
</table>
```

XML Element	Description Contains elements	Attributes	Description
<table>	Containing element for a table layout controlling its global functionality. - <rowId> - <column> - <rowPreview>	visible	Control table visibility with the following options: - always: The table always displays even if it is empty. - ifDatasourceAvailable (default): Display the table if an executable datasource is passed to it. - ifNotEmpty: Display the table if the datasource returns at least one row.
		selectionModel	Controls the row selection functionality with the following options: - none (default): No row selection is available. - one: Displays radio buttons for single row selection. - multiple: Displays checkboxes for multiple row selection. If selectionModel is one or multiple then element <rowId> is required.

		filter	Control column filter visibility with the following options: • none: No column filters. • hidden: Add a link allowing users to display filters. • auto (default): Display filters if there is more than one page of unfiltered data. • visible: Always display column filters.
		pageSize	Number of rows to display on a page (the default is 10).
<rowId>	Adds a datasource column with unique row identification which is required if the selectionModel attribute is set in the <table> element.	id	A unique row preview identifier.
<column>	Defines table columns and their content. • <content> • <property>	id	A unique column ID.
		label	An optional datasource column name used to sort the column.
		sortBy	An optional datasource column used to sort the column.
		sortable	An optional true (default) or false value to control if the column is sortable.
		filterBy	An optional datasource column name used to filter the column.

		filterable	An optional true or false value to control the visibility of the column filter.
		defaultSort Direction	An optional asc or desc value to mark the column as the primary column for sorting the table in the specified order.
<rowPreview>	Special definition to display a value across all columns, typically artifact descriptions. Users can enable/disable preview rows. • <content> • <property>	id	A unique row preview identifier.
<property>	Additional table content attributes.	name	A property name with options: • hidden: If true, the column is hidden with an option to show it using the column menu. • width: A number setting the column width. • fixed: If true, the column cannot be resized. • menuDisabled: If true, there is no column menu for the column. • visible: A true or false value for the visibility of a <rowPreview> element.
<content>	Defines the column content. There must be at least one content element in a column. • <parameter>	queryColumn	The name of the datasource column to display in the table column.
		componentName	An optional component identification used to define table content.

<parameter>	Optional name and value parameter pair to pass to a target component.	name	Parameter name with an optional constant value expressed as the parameter element content.
		queryColumn	Use this attribute to specify a value from a datasource column.

Table Actions Customization Reference

Actions define a set of functions displayed in the toolbar at the top of the table.

```
<actions>
  <actionComponent id="addDocumentArtifact" componentName="/core/documentation/uploadDocument"
         action="addDocumentArtifact" label="Upload"
         description="Upload a local file to the repository" styleClass="Upload">
    <parameter name="artifactType">DocumentationArtifact</parameter>
    <parameter name="artifact">${artifact}</parameter>
  </actionComponent>
  <actionComponent id="removeRelation" componentName="/core/relations/removeRelation"
         action="removeRelation" label="Delete" description="Delete selected items"
         styleClass="Remove" requiresSelection="true">
    <parameter name="relation">${relation}</parameter>
  </actionComponent>
</actions>
```

XML Element	Description Contains elements	Attributes	Description
<actions>	Containing element for a set of table actions. <actionComponent>		
<actionComponent>	A reference to a component that specifies a table action. <parameter>	id	A unique action ID.

XML Element	Description Contains elements	Attributes	Description
		componentName	The required component to render the action and its functionality.
		action	The action invoked when a user clicks the action button.
		label	A label for the action button.
		description	A tool tip for the action button.
		styleClass	An optional CSS class to associate with the button.
		requiresSelection	An optional true or false flag to declare that at least one table row must be selected to use this action.
<parameter>	Optional name and value parameter pairs to pass to the component.	name	Parameter name with the value expressed as the parameter element content.

Table Facets Customization Reference

Facets provides a set optional checkbox controls that enable table content filtering. Sets of values can come from some taxonomy or can also be a predefined list of values to display.

```
<facets>
  <facet id="serviceTypeFacet" label="Service type" queryColumn="a.r_serviceType" type="category">
    <parameter name="propertyName">a.r_serviceType</parameter>
  </facet>
  <facet id="stateFacet" label="State" queryColumn="stateString" type="category">
    <predefinedValue label="Public changes" value="PUBLIC_CHANGES"/>
    <predefinedValue label="Private changes" value="LOCAL_CHANGES"/>
    <parameter name="propertyName">state</parameter>
  </facet>
</facets>
```

XML Element	Description Contains elements	Attributes	Description
<facets>	Containing element for a set of facets. <facet>		
<facet>	A reference to a component that specifies a table action. • <parameter> • <predefinedValue>	id	A unique facet ID.
		label	A label for the facet.
		queryColumn	The name of the datasource column used to filter the result set. The values in this column must contain at least one checked facet value.
		type	An optional facet type for future development. The only valid value currently is `category`.
		taxonomy	An optional taxonomy URI used to obtain a list of possible values. Not required if you set a taxonomy property parameter or predefined values.
<parameter>	Additional facet parameters.	name	Set to `propertyName` and specify the SDM name of a taxonomy property used to obtain the values displayed in the facet.
<predefinedValue>	A user defined set of value to use in the facet.	label	A checkbox label
		value	The matching value in the query column if the box is checked.

Context Actions Customization Reference

The context actions component is a customizable set of context links on artifact detail pages. the customization must conform to `contextActions.xsd`.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="cust contextActions.xsd">
  <contextActions>
    <component componentName="/core/favorites/addFavorite" id="addFavorite">
      <parameter name="uuid">${artifact._uuid}</parameter>
    </component>
    <ruler id="ruler"/>
    <task id="navigator" label="Navigator" styleClass="UI PageAction Navigator"
                               taskLocation="/common/navigator">
      <parameter name="uuid">${artifact._uuid}</parameter>
    </task>
  </contextActions>
```

XML Element	Description • **Contains elements**	Attributes	Description
<contextActions>	Containing element for context actions. • <task> • <component> • <ruler>		
<task>	Defines a link to a framework task defined in the framework configuration file (`webui.xml`). <parameter>	id	Unique task identifier.
		label	The link label.
		taskLocation	The target task URI.
		styleClass	Optional CSS class to use for this link.

XML Element	Description • Contains elements	Attributes	Description
<component>	Includes a reference to the component which renders the context action. <parameter>	componentName	The component identification as defined in the web framework configuration file (`webui.xml`).
		id	A unique component placement identifier.
<ruler>	Display an item separator line.	id	A unique ruler identifier
<parameter>	Optional name and value parameter pairs to pass to the target task or component.	name	Parameter name with the value expressed as the parameter element content.

Query Selector Customization Reference

Query selectors define the filters used on browse pages. They render as a icon which corresponds to a particular query and also specify particular instances of search criteria and results tables.

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
    xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="cust queryList.xsd">
  <queries>
    <item icon="All" id="allServices" label="All" selected="true">
      <datasource>
        <query>select distinct :columns from businessServiceArtifact b left join
            hpsoaProjectArtifact p using p.r_projectContent</query>
      </datasource>
      <parameter name="criteria">criteria</parameter>
      <parameter name="table">results</parameter>
    </item>
    <item icon="Personal" id="myServices" label="My">
      <datasource>
```

```
        <query>select distinct :columns from businessServiceArtifact b (my) left join
            hpsoaProjectArtifact p using p.r_projectContent</query>
      </datasource>
      <parameter name="criteria">criteria</parameter>
      <parameter name="table">results</parameter>
    </item>
  </queries>
</customization>
```

XML Element	**Description** - **Contains elements**	**Attributes**	**Description**
<queries>	Containing element for query selectors. <item>		
<item>	Defines a particular query filter with a corresponding set of criteria and a results table. - <datasource> - <parameter>	icon	Optional CSS classname to render the filter icon.
		id	A unique query filter ID.
		label	The display label for the query filter.
		selected	Optional true or false to mark the query filter as the default when a user opens the page.

XML Element	Description • Contains elements	Attributes	Description
<datasource>	Containing element for the datasource defining the query. • <type> • <query> • <parameter>		
<type>	An optional element which defaults to `dql`.		
<query>	A shortcut for <parameter name="query"> containing the actual query.		
<parameter>	Within <item> there are two required parameters specifying the IDs of the matching criteria and table customizations to match the filter. For <datasource> the parameter is an optional replacement for an element in the query.	name	Parameter name with the value expressed as the parameter element content. `name="criteria"` and `name="table"` are required within the <item> element. The value appends to the item ID to specify the Id of these customizations.

Search Criteria Customization Reference

Search criteria define the options available in the search component on browse pages. Each browse page may have multiple sets of criteria, each corresponding to a defined filter. Select the filter first to ensure that you customize the correct criteria.

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="cust searchCriteria.xsd">
  <search>
    <criteria id="nameAndDescription" label="Query" type="properties" visible="true"
              widget="search:nameAndDescription">
```

```
      <parameter name="name">b.name</parameter>
      <parameter name="name">b._longDescription</parameter>
    </criteria>
    <criteria id="keywords" label="Keywords" type="property" visible="true">
      <parameter name="name">b.keyword</parameter>
    </criteria>
    <criteria id="provider" label="Provider" type="relation" widget="text">
      <parameter name="name">d.name</parameter>
      <parameter name="relation">d.provides</parameter>
      <parameter name="targetSdmName">contactArtifact</parameter>
      <parameter name="useTypes">provider;EMPTY</parameter>
    </criteria>
    <criteria id="artifactType" label="Artifact Type">
      <parameter name="name">consumer.artifactType</parameter>
      <widgetParameters>
         <parameter name="availableArtifactTypes">${availableArtifactTypes}</parameter>
      </widgetParameters>
    </criteria>
  </search>
</customization>
```

XML Element	**Description** • **Contains elements**	Attributes	Description
<search>	Containing element for search criteria. • <datasource> • <criteria>		

XML Element	Description - **Contains elements**	Attributes	Description
<datasource>	Optional datasource to define the datasource specifically for the search criteria. The datasource is normally passed from the query selector instead. - <type> - <query> - <parameter>		
<criteria>	Defines a particular search criteria. - <parameter> - <widgetParameters>	id	A unique criteria ID.
		label	A criteria display label.
		type	The type of criteria from the following options: - **properties**: Filter over multiple properties. The requested value must match at least one specified artifact property to generate a result. - **property**: Filter by a single primitive or taxonomy property. - **relation**: Filter by a property of a related artifact. - **custom**: Use a special filter (for example, artifact type).

XML Element	Description • Contains elements	Attributes	Description
		visible	If true, the criteria is visible by default, if false, the criteria is hidden but available using the **Add Criteria** function.
		widget	The name of the widget to display criteria input. Typically it is obtained automatically from the property name.
<widgetParameter>	Parameters passed to a widget component responsible for criteria rendering (for example, taxonomy identification to obtain a list of categories). <parameter>		
<parameter>	Parameter name and value pairs to pass to the datasource filter created by the criteria or the widget component used to render the criteria.	name	The parameter name with the value expressed as the element content.

Search Component Customization Reference

The search component on home pages displays an input field and returns dynamic results as you type.

```
<?xml version="1.0" encoding="UTF-8"?>
<customization xmlns="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:cust="http://soa.systinet.hp.com/2009/02/ui/customization"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="cust suggestedSearch.xsd">
  <suggestConfig minCharacters="2" typedelay="500">
    <datasource>
      <query>select :columns from artifactBase a</query>
    </datasource>
    <search>
      <criteria id="nameAndDescription" label="" type="properties" visible="true"
```

```
                 widget="search:nameAndDescription">
        <parameter name="name">name</parameter>
        <parameter name="name">_longDescription</parameter>
      </criteria>
    </search>
    <table>
      <column id="id" label="Uuid">
        <content queryColumn="a._uuid"/>
        <property name="hidden">true</property>
      </column>
      <column id="name" label="">
        <content queryColumn="a.name"/>
        <property name="css">Name</property>
      </column>
      <column id="version" label="Version">
        <content queryColumn="a.version"/>
        <property name="css">Version</property>
      </column>
    </table>
    <actionComponent action="viewArtifact" componentName="/core/search/searchRedirectToArtifact"
             id="viewArtifactRedirect" label="OnSelect">
      <parameter name="id">${id}</parameter>
    </actionComponent>
  </suggestConfig>
</customization>
```

XML Element	Description Contains elements	Attributes	Description
<suggestConfig>	Containing element for dynamic search. • <datasource> • <search> • <actionComponent>	minCharacters	The minimum number of characters the user types before dynamic search starts.
		typeDelay	The delay between the last key press and starting dynamic search.
<datasource>	The datasource to define the search. • <type> • <query> • <parameter>		
<search>	Defines the filtering criteria for matching the input to the query set. <criteria>		
<criteria>	Defines a particular search criteria. <parameter>	id	A unique criteria ID.
		label	A criteria display label.

XML Element	Description Contains elements	Attributes	Description
		type	The type of criteria from the following options: - **properties**: Filter over multiple properties. The requested value must match at least one specified artifact property to generate a result. - **property**: Filter by a single primitive or taxonomy property. - **relation**: Filter by a property of a related artifact. - **custom**: Use a special filter (for example, artifact type).
		visible	If true, the criteria is visible by default, if false, the criteria is hidden but available using the **Add Criteria** function
		widget	The name of the widget to display criteria input. Typically it is obtained automatically from the property name.
<table>	Defines a description for the dynamic result display. <column>		
<column>	Defines a column of the dynamic results specifying a property to display. - <content> - <property>	id	A unique column ID.
		label	A column display label.

XML Element	Description Contains elements	Attributes	Description
<content>	Defines the column content.	queryColumn	The query column to display.
<property>	Additional properties for the column.	name	Specify the CSS class to render the column.
<actionComponent>	Specify the action when a user clicks a dynamic result. • <parameter>	componentName	Specify the component to call on user selection.
		action	The action to trigger with this component.
		id	A unique ID.
		label	the name of the client browser the component listens to, typically `OnSelect`.
<parameter>	Additional action or criteria parameter name and value pairs to pass to the datasource filter created by the criteria or the action component.	name	The parameter name with the value expressed as the element content.

For more details about DQL queries, see "Using DQL" in the *Developer Guide*.

HTML Areas Customization Reference

These components are not customized directly using XML in the editor like other components. Instead. Systinet provides a rich-text HTML editor.

You can edit rich text using the Editor toolbar buttons, or switch to source HTML editing with the 📝 button.

Systinet allows the following HTML tags and attributes:

HTML Tag	Description	Attributes
a	Anchor, often a URL.	href
b	Bold text.	

HTML Tag	Description	Attributes
br	Single line break.	
em	Emphasized text.	
font	Font, color, and size for text.	face
		size
		color
i	Italic text.	
li	List item.	
ol	Ordered List.	
p	Paragraph.	
strong	Strong text.	
u	Underlined text.	
ul	Unordered list.	

Note: For security reasons only a very limited set of markup is allowed, all other disallowed markup is filtered out when you save.

Systinet stores the data in XML format according to the `contentWrapper.xsd` schema.

```
<content>
<![CDATA[
<p>
Welcome to HP SOA Systinet service catalog. You can find information about existing applications,
services and processes available for reuse here, bookmark them using the 'Favorites' function,
create new, attach business requirements and request contracts.
</p>
]]>
```

```
</content>
```

The content is represented in a CDATA section because it does not need to be valid XML (for example, <br> elements).

Reusable UI Components Reference

The following components are available for use in various places in customization XMLs. You can include them using the component element with a proper `componentName` attribute (for example, in main page layouts or as table cell content). For each component there is a description, the elements in the UI where you can use them and list of all required and optional parameters.

Systinet provides the following components organized into the UI elements where you can use them:

- “Page Content Components”
- “Column Components”
- “Artifact Property Components”
- “Left Menu Components”
- “Table Action Components”
- “Table Cell Components”
- “Context Action Components”

Page Content Components

Component Name **Description**	**Parameter**	**Parameter Description**
/core/browse/browseArtifactCriteria Provides a customizable set of additional browse criteria.	customizationId (required)	Specify the id of the customization component.

Page Content Components, continued

Component Name Description	Parameter	Parameter Description
	datasource (required)	The datasource for user-defined criteria. You must use the same instance in the result table to display the query results.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.
/core/contact/manageContacts Allow users to view or edit artifact contacts grouped by job roles.	artifactUuid (required)	The UUID of the artifact to manage contacts for.
/core/relations/groupingReadonlyRelationshipComponent Allows user in grouping and enabling readonly relationship functionality.	groupNames (required)	The labels in the table is from the 'groupNames' parameter.
	groupConditions (required)	SDM properties defined in 'groupCondition' parameter must be defined in 'properties' parameter. Each condition is separated by comma and is displayed by individual table.
	artifact (required)	The instance of artifact to readonly.
	relation (required)	The SDM name of the relationship which targets the artifact type in the table.
	artifactType (required)	The type of artifact for grouping readonly components.

Page Content Components, continued

Component Name Description	Parameter	Parameter Description
	properties (required)	Display the properties of the artifacts.
	displayColumns (required)	Display the columns in the table.
	useTypeColumn (required)	Use the artifacts of the specified type.
/core/eventlog/eventPreview Displays a list of recent user events. For example, contract request changes.	messageCount (required)	Maximum number of events to show. A list of all user events is accessible with the **More** link.
	limitToDay (required)	Do not show events older than the specified number of days for all events pages.
	messageTypes (optional)	Specify the types of message to display. Event types are specified by event log providers (for example, 10001 is approval request events).
/core/groupMembership Display a list of groups that the current user is a member of.	artifact (required)	The instance of the Person artifact for the user.
/core/layout/header Display a page content header with optional subtitle and icon.	title (optional)	Specify a page title, defaults to the current task caption.
	subtitle (optional)	Specify a page subtitle. Empty by default.
	subtitleClass (optional)	CSS class used to render the subtitle. A possible value is `Artifact`.

Page Content Components, continued

Component Name Description	Parameter	Parameter Description
	iconType (optional)	The name of the icon to display, typically an artifact type (for example, businessServiceArtifact).
	iconFormat (optional)	Specify the version of the icon to display from options small, large, or vector.
/core/relations/genericRelationshipComponent Display a table of related artifacts and enable add/remove relationship functionality.	tableDefinitionId (required)	Specify a customization ID for the table.
	artifact (required)	The instance of ArtifactBase for the current artifact.
	relation (required)	The SDM name of the relationship which targets the artifact type in the table.
	useType (optional)	Only display relationships with this useType. Also set for newly created relationships
	addMode (optional)	Specify how the user selects target artifacts with options, search (default) and browse.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.

Page Content Components, continued

Component Name Description	Parameter	Parameter Description
/core/search/homepageSearch Provide a search component with dynamic results and link to advanced search.	customizationId (required)	Specify the ID of this custom component.
	viewId (required)	Specify which tab to open when the user clicks **Go**.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.
/core/search/searchCriteria Provide a customizable set of additional criteria.	customizationId (required)	Specify the ID of this custom component.
	datasource (required)	The datasource containing user-defined criteria. you must use the same instance in the results table to display the query results.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.

Page Content Components, continued

Component Name Description	Parameter	Parameter Description
/core/search/searchQueryList Provide a customizable list of queries for browse pages.	customizationId (required)	Specify the ID of this custom component.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.
/core/table Display data from a datasource in a table, with sorting, paging, and filtering.	customizationId (required)	Specify the ID of this custom component.
	datasource (optional)	The datasource providing the query results for the table. If it is not provided as a parameter, it must be defined in the table.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.
/core/taxonomy/homepage/homepageView Display a list of taxonomies with categorized artifact counts and links to category browsing.	homepageGid (required)	Identify the reduced set of taxonomies to display in this component. The list of taxonomies is available in `ui.taxonomyGroupsPreferences.xml` in `SYSTINET_HOME/conf/setup/data`.

Page Content Components, continued

Component Name **Description**	**Parameter**	**Parameter Description**
	allGid (required)	Identify the group of full taxonomy lists to display in the Category Browsing page accessible with the **More** link.
/core/text/editableRichTextArea Display a customizable HTML text area.	customizationId (required)	Specify the ID of this custom component.
	title (optional)	Text area title text.
/detail/editArtifactProperties Display customizable artifact properties or other artifact related components in editable mode.	customizationId (required)	Specify the ID of this custom component.
	artifact (optional)	The instance of Artifact Base to edit.
	subtitle (optional)	Text to display in the page content header.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.
/detail/viewArtifactProperties Display customizable artifact properties or other artifact related components in view mode.	customizationId (required)	Specify the ID for this custom component.
	artifact (required)	The instance of ArtifactBase to view.
	subtitle (optional)	Text to display in the page content header.

Page Content Components, continued

Component Name Description	Parameter	Parameter Description
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.

Column Components

Component Name Description	Parameter	Parameter Description
/core/contact/contactsMenu Renders a list of contacts ordered by their useTypes.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
	readOnly (optional)	If true, then link from the current component to a read-only contact management page.
	maximumVisibleItems (optional)	Specify the maximum number of contacts to display. The default is to display all of them.
/core/contract/contractMenu Renders a list of contracts associated with the current artifact.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
	viewId (required)	If viewId is analyst, shows all user contracts, if viewId is partner, shows only contract requests for the current user.
	maximumVisibleItems (optional)	Specify the maximum number of contracts to display. A full list is available with the **More** link.

Column Components, continued

Component Name Description	Parameter	Parameter Description
/core/contract/myContracts Renders a list of current accepted and pending contracts for the current user.	numberOfContracts (optional)	Specify the maximum number of contracts to display. A full list is available with the **More** link.
/core/favorites/favoritesList Display a list of favorite artifacts for the current user.	count (optional)	Specify the number of favorites to show (the default is 5).
/core/layout/contextActions Display customizable context actions for the current artifact.	customizationId (required)	Specify the Id of this customization component.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.

Artifact Property Components

Component Name Description	Parameter	Parameter Description
/core/contact/contractProvider Renders a provider property in contract or contract request artifact detail pages.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
/core/contact/providerWrapper Renders a virtual Provider property in artifact details and provides a dialog to change it.	customizationId (required)	Specify the ID of this customization component.

Artifact Property Components, continued

Component Name **Description**	**Parameter**	**Parameter Description**
	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
	readOnly (optional)	If true, then users cannot change the provider.
	wrapCaption (optional)	If provided, the whole component is surrounded by a color area with the specified title.
	wrapStyle (optional)	If wrapCaption is specified, use wrapStyle to specify the style of the component outline with options, `Common` (default) and `Green`.
/core/util/parentLinksRenderer Display links to parents of the current artifact.	uuid (required)	The UUID of the current artifact.
	sdmName (required)	The SDM name of the current artifact..
	maxCount (optional)	The maximum number of parent artifact to show (default is 3).
/core/util/providerLinksRenderer Display links to providers of the current artifact.	uuid (required)	The UUID of the current artifact.
	sdmName (required)	The SDM name of the current artifact..
	maxCount (optional)	The maximum number of parent artifact to show (default is 3).

Left Menu Components

Component Name	Description	Parameter
/core/recentlyUsedArtifacts	Display a list of the last 5 artifact the user viewed or edited .	None

Table Action Components

Component Name Description	Parameter	Parameter Description
/core/contact/AddContact Table action to add a relationship between the current artifact and a selectable contact artifact.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
	useType (required)	The contact role from the Job Roles taxonomy.
/core/documentation/uploadDocument Open a document upload pop-up window. Uploaded documents are associated with the current artifact.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
/core/relations/addNew Table action to create a new artifact and the relationship to the current artifact.	artifact (required)	The instance of ArtifactBase for the current artifact.
	artifactSdmName (required)	The SDM name of the artifact type to create
	relation (required)	The SDM name of the relationship to use between the current artifact and the newly created one.
/core/relations/addRelation Create a relationship of the specified type between the current artifact and a user selected artifact.	relation (required)	The SDM name of the relationship to use between the current artifact and the selected one.

Table Action Components, continued

Component Name Description	Parameter	Parameter Description
	useType (optional)	An attribute used to render newly created relationships.
/core/relations/removeRelation Removes relationships between the selected artifacts and the current artifact.	relation (required)	The SDM name of the relationship to remove.
	useType (optional)	Remove only relationships of the specified type.

Table Cell Components

Component Name Description	Parameter	Parameter Description
/core/dowloadWsdl Display a link to the document for a specific environment.	uuid (required)	The UUID of the WSDL artifact.
	implUuid (optional)	The UUID of the implementation artifact.
	selectedEnv (optional)	The requested environment name, must be used with the implUuid parameter.
/core/favorites/addFavorite Display a clickable icon in a table cell. The current artifact status is indicated by the icon color.	uuid (required)	The UUID of the current artifact.
/core/util/parentLinksRenderer Display links to parents of the current artifact.	uuid (required)	The UUID of the current artifact.

Table Cell Components, continued

Component Name **Description**	**Parameter**	**Parameter Description**
	sdmName (required)	The SDM name of the current artifact.
	maxCount (optional)	The maximum number of parent artifact to show (default is 3).
/core/util/providerLinksRenderer Display links to providers of the current artifact.	uuid (required)	The UUID of the current artifact.
	sdmName (required)	The SDM name of the current artifact.
	maxCount (optional)	The maximum number of parent artifact to show (default is 3).

Context Action Components

Component Name **Description**	**Parameter**	**Parameter Description**
/core/contextActions/deleteArtifact Renders a delete artifact context link.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
/core/contextActions/editArtifact Renders an edit artifact context action.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.
/core/contract/action/contractRequestWizard Renders a contract request wizard link. Only displayed for artifacts which are *Ready for Consumption*.	artifact (required)	The instance of the ArtifactBase object representing the current artifact.

Reusable UI Tasks Reference

The following table lists the task URIs and parameters you can use to reference pages in role-based UIs.

Tasks prefixed by `/common` keep the current UI active. If you want to specify a page in a specific tab replace `/common` with the tab identification, for example, `/analyst/browseArtifact` instead of `/common/browseArtifact`. There is no common homepage, so you must use specific tab IDs.

Task URI	Description	Parameter	Description
/common/advancedSearch	The advanced search page, it prepares an empty datasource usable by criteria and table components.		
/common/browseArtifact	The browse page for the specified artifact type. The artifact type must be defined as a primary artifact for the active tab.	artifactType (required)	The SDM name of the artifact type to display.
		title (optional)	The text to display for the browse page header.
/common/categorization/categoryView	Entry point for the category browser.	allGid (required)	Defines the id of the taxonomy group containing all visible taxonomies in the category browser. The definition of taxonomy groups is in `ui.taxonomyGroupsPreferences.xml` in `SYSTINET_HOME/conf/setup/data/`.
/common/editArtifact	Opens the edit page for the specified artifact, artifact type must be listed in the active tab as a primary or secondary artifacts.	uuid (required)	The uuid of the artifact to edit.

Task URI	Description	Parameter	Description
/common/favorites	Open the Favorites page for the current user.		
/common/newArtifact	Open the new artifact page for the specified artifact type.	sdmName (required)	The SDM name for the artifact type to create.
/common/contract/contractRequestWizard	Open the contract request wizard.	uuid (required)	The uuid of the provider artifact.
/[tabID]/homepage	Open the homepage for the specified tab.		
/artifact/[artifactUuid]	Open the artifact detail page for the specified artifact.		
/document/[artifactUuid]	Open the attached data content for the specified artifact. For example, the actual document associated with a Documentation artifact.		

Tab Configuration Reference

Each UI tab is described by an XML configuration file, which contains basic settings, for example, the user perspective required to see this tab, the homepage link, and the visible artifact types.

```
<uiConfiguration>
  <id>partner</id>
  <label>Partner UI</label>
  <perspective>businessPartner</perspective>
  <task>/partner/homepage</task>
  <weight>200</weight>
  <documentationLink>doc/html/index.html?bpuser.html</documentationLink>
  <approvedDataOnly>true</approvedDataOnly>
  <primaryArtifact>businessServiceArtifact</primaryArtifact>
  <secondaryArtifact>contractArtifact</secondaryArtifact>
  <secondaryArtifact>contractRequestArtifact</secondaryArtifact>
  <secondaryArtifact>contactArtifact</secondaryArtifact>
  <secondaryArtifact>personArtifact</secondaryArtifact>
  <secondaryArtifact>organizationUnitArtifact</secondaryArtifact>
  <secondaryArtifact>sloArtifact</secondaryArtifact>
  <secondaryArtifact>documentationArtifact</secondaryArtifact>
  <taxonomyGroup>8</taxonomyGroup>
  <artifactType sdmName="contactArtifact">
  <taxonomy>uddi:systinet.com:soa:model:taxonomies:
       contactRoles</taxonomy>
  <taxonomy>uddi:systinet.com:soa:model:taxonomies:
       contactTypes</taxonomy>
  <taxonomy>uddi:systinet.com:soa:model:taxonomies:
       contactClassifications</taxonomy>
  </artifactType>
</uiConfiguration>
```

Tab configurations contain the following elements:

XML Element	Description
id	Unique tab id.
label	Label displayed in the UI tab.
perspective	The perspective associated with the tab. Note that users associated with a role-based UI cannot see the standard UI.
task	The tab homepage task URI.
weight	Optional number for tab ordering if more than one tab is visible to a perspective.
documentationLink	Optional link to documentation for the tab.
approvedDataOnly	If true, then only approved revisions of artifacts appear in the tab, if false then latest revisions appear.
primaryArtifact	A list of artifact SDM names available for browsing and for display in the tab.

XML Element	Description
secondaryArtifact	A list of artifact SDM names available for display in the tab.
taxonomyGroup	Contains the ID of the taxonomy group used where lists of categories are visible.
artifactType	Specifies additional taxonomies visible in the Categories UI component with the following attributes and elements: • sdmName - the SDM name of the artifact type in which the taxonomy is visible. • taxonomy - the URNs of the additional taxonomies.

Customizing Dependency Impact Report

The composite browser do not satisfy some of the requirements. Therefore there is a customizable component that accepts a variable configurations and produces result in a standard UI table.

Customizing Dependency/Impact Report Design

> **Note:** Customization options are only available to users in the HP SystinetAdministrator role.

The following demonstration will show you how to implement your own design.

Adding a table to the overview tab:

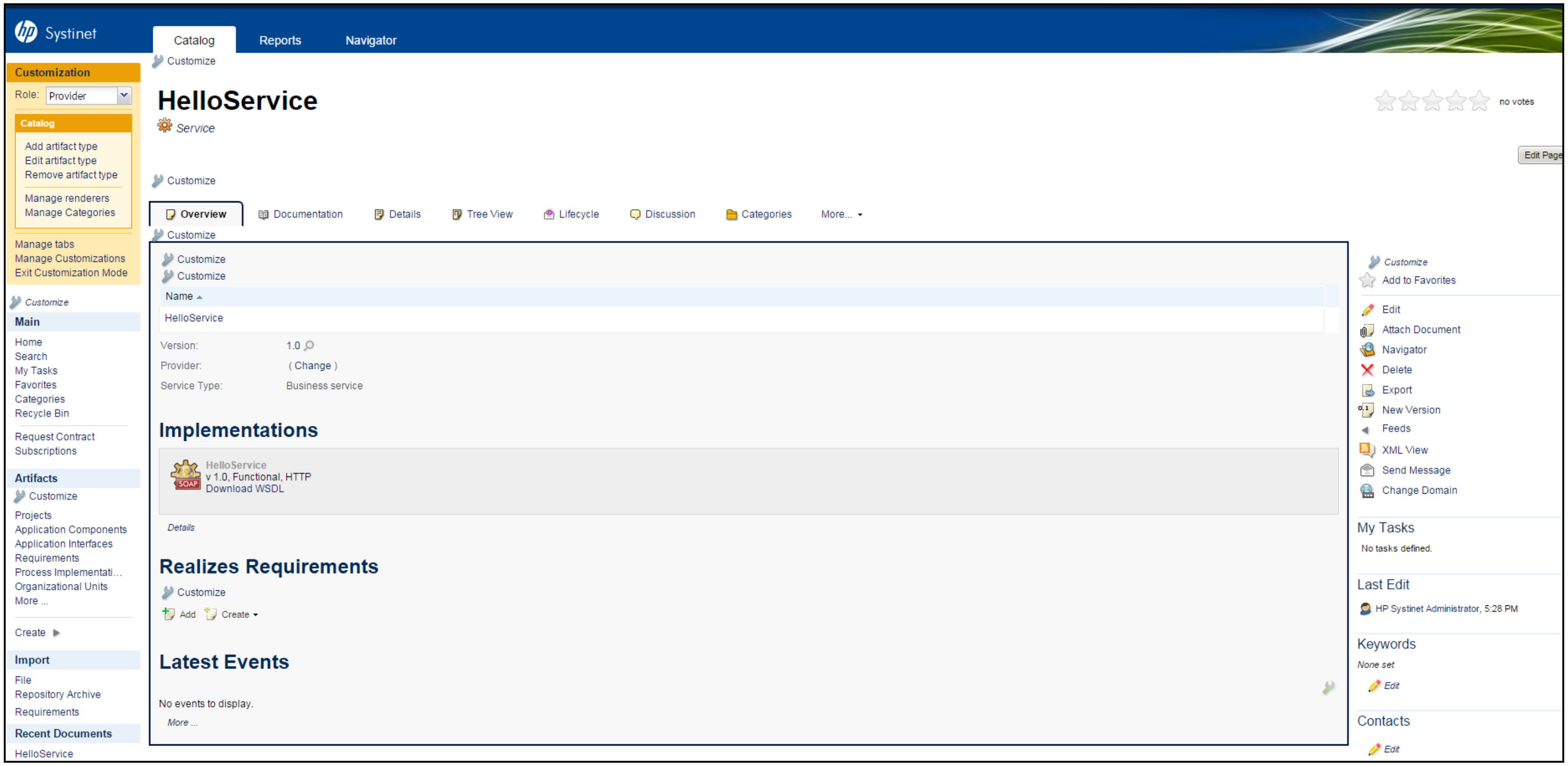
1. Switch to customization mode and navigate to a business service. See "UI Customization" in the *Concepts Guide* for details.

2. Add the following code:

```
<?xml version="1.0" encoding="UTF-8"?>

<customization xmlns="http://systinet.hp.com/2009/02/ui/customization"
xmlns:cust="http://systinet.hp.com/2009/02/ui/customization" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="cust columns.xsd">

   <columns>
```

```
      <column id="leftcolumn">

            <!-- THE FOLLOWING NEEDS TO BE ADDED -->

            <component componentName="/core/table" id="table">

               <parameter
name="customizationId">provider.viewArtifact.businessServiceArtifact.impact.table</parameter>

            </component>

            <!-- END OF INCLUDE -->

      ....
```

Resulting in this:



Version: 1.0

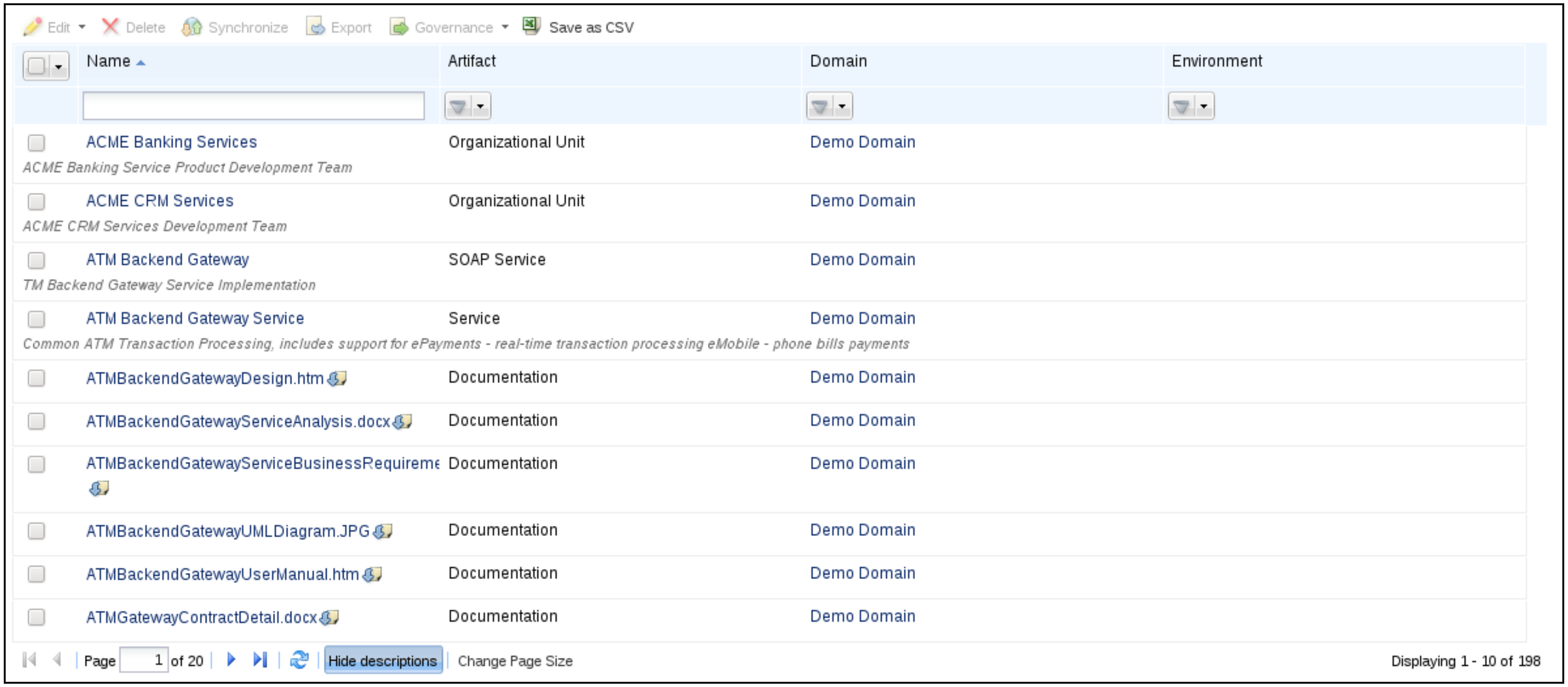
Provider: (Change)

Service Type: Business service

3. Click on the Customize link just above the table and replace the definition with the following:

```
<?xml version="1.0" encoding="UTF-8"?>

<customization xmlns="http://systinet.hp.com/2009/02/ui/customization"
xmlns:cust="http://systinet.hp.com/2009/02/ui/customization" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="cust table.xsd">

   <datasource>

      <type>dataSource.composite.report</type>

      <parameter name="seed">${artifact._uuid}</parameter>

      <parameter name="configuration"><![CDATA[
         <closure maxDepth="20" maxResults="1000" maxProcessingTime="60000" debug="true">
            <resultArtifacts>
               <artifact sdmName="artifactBase"/>
               <artifact sdmName="endpointArtifact">
                  <field name="environment" property="environment.val"/>
               </artifact>
            </resultArtifacts>
         </closure>
      </parameter>
   </datasource>
   <table selectionModel="multiple">
      <rowId queryColumn="id"/>
      <column id="name" label="Name">
         <content queryColumn="name"/>
      </column>
      <column id="type" label="Artifact">
         <content queryColumn="_sdmName"/>
      </column>
      <column id="_domainName" label="Domain">
         <content queryColumn="_domainName"/>
      </column>
      <!-- This column is user defined and matches the environment field within the endpoint result artifact -->
      <column id="environment" label="Environment">
         <content queryColumn="environment"/>
      </column>

      <rowPreview id="description">

         <content queryColumn="description"/>

      </rowPreview>

   </table>

   <actions disableExports="false">

   </actions>

</customization>
```

Resulting in this:



	Name	Artifact	Domain	Environment
□	ACME Banking Services *ACME Banking Service Product Development Team*	Organizational Unit	Demo Domain	
□	ACME CRM Services *ACME CRM Services Development Team*	Organizational Unit	Demo Domain	
□	ATM Backend Gateway *TM Backend Gateway Service Implementation*	SOAP Service	Demo Domain	
□	ATM Backend Gateway Service *Common ATM Transaction Processing, includes support for ePayments - real-time transaction processing eMobile - phone bills payments*	Service	Demo Domain	
□	ATMBackendGatewayDesign.htm	Documentation	Demo Domain	
□	ATMBackendGatewayServiceAnalysis.docx	Documentation	Demo Domain	
□	ATMBackendGatewayServiceBusinessRequireme	Documentation	Demo Domain	
□	ATMBackendGatewayUMLDiagram.JPG	Documentation	Demo Domain	
□	ATMBackendGatewayUserManual.htm	Documentation	Demo Domain	
□	ATMGatewayContractDetail.docx	Documentation	Demo Domain	

See "Datasource Configuration" on the next page for more parameter details.

Datasource Configuration

The datasource has two basic parameters:

The traversal rules specified with a XML configuration (see the `<closure>` tag for reference).

The seed parameter of `seedQuery` specifying which artifacts the impact report is created for.

seed parameter

The seed parameter specifies the uuid of the artifact you are interested in.

`seedQuery` parameter

The `seedQuery` parameter specifies a dql query which is expected to return a result set of with one column returning a list of uuids.

`<closure>`

The wrapping element of the configuration. It defines the following:

Name	Type	Default Value	Description
maxDepth	attribute	5	maximum distance of the result from the seed artifact specified in the number of traversed relationships
maxResults	attribute	20060000	the report processing thread quits after producing specified number of results
maxProcessingTime	attribute	0	the report processing thread quits it's operation after the specified time in miliseconds
nice	attribute	true	the report processing thread sleeps every 100 processed results for the given number of miliseconds. Expected to be used for longer running reports that may jam the server for other users
debug	attribute	required	when true, detailed tracing information is written into the log file
resultArtifacts	element	N/A	artifact that form the result

Name	Type	Default Value	Description
traversableArtifacts	element	N/A	artifacts that can be walked through when creating the report. Only one of traversableArtifacts,nontraversableArtifacts can be specified.
nonTraversableArtifacts	element	N/A	artifacts that can't be walked through when creating the report. Only one of traversableArtifacts,nontraversableArtifacts can be specified.
traversableRelations	element	N/A	relationships that can be walked through when creating the report. Only one of traversableRelations,relationStopList can be specified.
relationStopList	element	N/A	relationships that can't be walked through when creating the report. Only one of traversableRelations,relationStopList can be specified.

- You can combine the `traversableArtifact` section with `traversableRelations` or `relationStopList`. If you do so, conditions of both settings will be applied. In the same manner you can combine `nonTraversableArtifacts`.
- Result artifacts are not traversable by default; they are added to results when reached according to specified rules. If you want to traverse relationships leading from these artifacts you have to add them to the list of traversable artifacts.

`<artifact>`

The list of artifact types that form the results. Each artifact result type may define a set of fields that will form the result. (`name,description,domainId` are the automatically added default fields)

```
<artifact sdmName="businessServiceArtifact" query="from businessServiceArtifact a where a.consumable='1' and a._
uuid=:uuid">

   <field name="implementationCount" query="select count(i._uuid) from businessServiceArtifact b join
implementationArtifact i using service where b._uuid=:uuid"/>

</artifact>
```

- Using the optional query attribute you can filter matching artifact instances. The 'artifact' tag can be nested within `traversableArtifacts`,`nonTraversableArtifacts` or `resultArtifacts`. The artifact instance is matched when the query returns at least 1 result. You need to utilize the 'uuid' parameter in the query which holds the artifact uuid which is subject to the matching.

- You can use abstract artifacts in the place of `sdmName`. In that case the rule will apply to all artifact which extend the specified artifact in addition.

- You can use optional `reachedUsing` attribute which can filter traversed artifacts based on the relationship these have been reached. There are the following options of the value of the attribute:

incoming	The traversed artifact will be treated by the engine only if it was reached over an incoming relationship. The artifact will be treated as non existing otherwise.
outgoing	The traversed artifact will be treated by the engine only if it was reached over an outgoing relationship. The artifact will be treated as non existing otherwise.
comma separated list of relationships	The traversed artifact will be treated by the engine only if it was reached over relationship which sdm name is present within one of the values of the list defined by this attribute. The artifact will be treated as non existing otherwise.

In the following example the the report is launched from the endpoint artifact (which is linked to a `webServiceArtifact`) It will traverse through the `webServiceArtifact` using the `endpointOf relationship` (which is an incoming relationship inside SDM). If you would change the value of `reachedUsing` to 'incoming', the traversal through `webServiceArtifact` would happen as well. If you would change it to 'outgoing' you would not get any results.

```
<closure maxDepth="20" maxResults="1000" maxProcessingTime="60000" debug="true">

   <resultArtifacts>

      <artifact sdmName="businessServiceArtifact"/>

      <artifact sdmName="endpointArtifact"/>

   </resultArtifacts>

   <traversableArtifacts>

      <artifact reachedUsing="endpointOf" sdmName="implementationArtifact"/>

   </traversableArtifacts>

</closure>
```

<field>

The field element specifies an extra field in the result row and can be used as child of the `artifact` tag within `resultArtifacts`. There are two methods how to specify the field:

1. Via DQL query

```
<field name="implementationCount" query="select count(i._uuid) from businessServiceArtifact b join
implementationArtifact i using service where b._uuid=:uuid"/>
```

2. Via property sdm name specification

```
<field name="consumable" property="consumable"/>
```

The second variant has much better performance and should be used wherever possible.

Note: To make the field visible you have to add an extra column to the table definition and link the field to it.See the examples with the environment property.

There are several predefined fields (artifact fields) that you don't need to define explicitly:

_domainId
_domainName
_owner
name
description

For more information, see the DQL or SDM model documentation.

`<relation>`

Specifies a relation and can be used with `relationStopList` tag and `traversableRelations` tag.

```
<relation sdmName="r_dependsOn"/>
```

`<shortestPath>`

Specifies all relations defining the shortest path between the source and target artifact within the SDM model. Relations that have a range of type property groups can not be part of the resulting path. This construct is intended to ease the definition of more complex configurations and combining with the SDM configuration. For instance, when an endpoint and `businessService` is specified, it will find the way through the implementations. It is good to verify in the debug mode that the shortest path is really what you really want.

Can be used with `relationStopList` tag and `traversableRelations` tag.

```
<shortestPath sourceArtifact="businessServiceArtifact" targetArtifact="endpointArtifact"/>
```

Examples

All examples are expected to be used within the configuration provided with the Hello World sample. Replace the data source configuration with the provided example. All the screen shots are made with demo data showing the "*ATM Backend Gateway Service*" business service.

Example 1

List all reachable artifacts from the seed artifact:

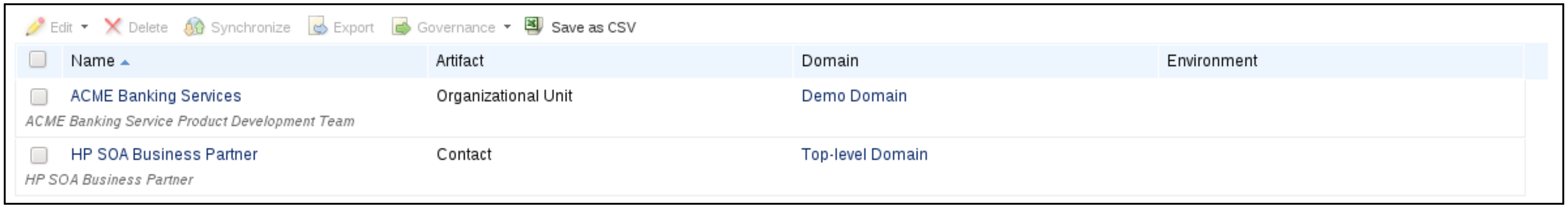
```
<closure maxDepth="5" maxResults="1000" maxProcessingTime="60000" debug="true">

   <resultArtifacts>

      <artifact sdmName="artifactBase"/>

   </resultArtifacts>

</closure>
```

Example 2

Show all contacts having a contract on the seed business service:

```
<closure maxDepth="5" maxResults="100" maxProcessingTime="60000" debug="false">

   <resultArtifacts>

      <artifact sdmName="contactArtifact"/>

   </resultArtifacts>

   <traversableArtifacts>

      <artifact sdmName="contractArtifact" query="from contractArtifact a where a._uuid=:uuid and a.contractState.val
= 'uddi:systinet.com:soa:model:taxonomies:contractAgreementStates:accepted'"/>

   </traversableArtifacts>

</closure>
```

Resulting in:



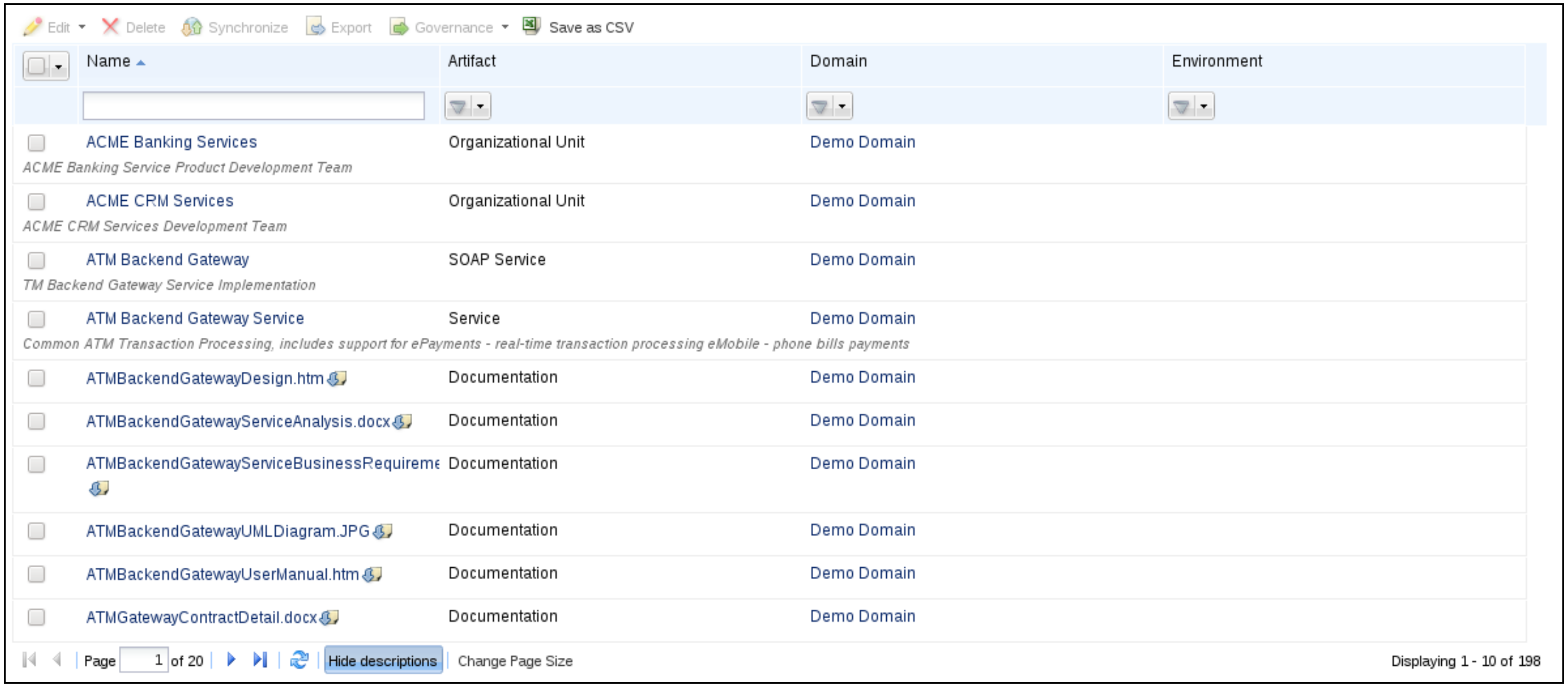
□	Name	Artifact	Domain	Environment
□	ACME Banking Services ACME Banking Service Product Development Team	Organizational Unit	Demo Domain	
□	HP SOA Business Partner HP SOA Business Partner	Contact	Top-level Domain	

Example 3

Show all endpoints provided by a business service through all its implementations:

```
<closure maxDepth="5" maxResults="100" maxProcessingTime="60000">

   <resultArtifacts>

      <artifact sdmName="endpointArtifact">

         <field name="environment" property="environment.val"/>

      </artifact>

   </resultArtifacts>

   <traversableRelations>

      <shortestPath sourceArtifact="businessServiceArtifact" targetArtifact="endpointArtifact"/>

   </traversableRelations>

</closure>
```

Resulting in:



□	Name	Artifact	Domain	Environment
□	ACME Banking Services *ACME Banking Service Product Development Team*	Organizational Unit	Demo Domain	
□	ACME CRM Services *ACME CRM Services Development Team*	Organizational Unit	Demo Domain	
□	ATM Backend Gateway *TM Backend Gateway Service Implementation*	SOAP Service	Demo Domain	
□	ATM Backend Gateway Service *Common ATM Transaction Processing, includes support for ePayments - real-time transaction processing eMobile - phone bills payments*	Service	Demo Domain	
□	ATMBackendGatewayDesign.htm	Documentation	Demo Domain	
□	ATMBackendGatewayServiceAnalysis.docx	Documentation	Demo Domain	
□	ATMBackendGatewayServiceBusinessRequireme	Documentation	Demo Domain	
□	ATMBackendGatewayUMLDiagram.JPG	Documentation	Demo Domain	
□	ATMBackendGatewayUserManual.htm	Documentation	Demo Domain	
□	ATMGatewayContractDetail.docx	Documentation	Demo Domain	

Page 1 of 20 | Hide descriptions | Change Page Size

Displaying 1 - 10 of 198

Example 4

Show all business services that the seed one is transitively referencing using the relationship `r_dependsOn`:

```
<closure maxDepth="5" maxResults="100" maxProcessingTime="60000" debug="false">
   <resultArtifacts>
      <artifact sdmName="businessServiceArtifact"/>
   </resultArtifacts>
   <traversableRelations>
      <relation sdmName="r_dependsOn"/>
   </traversableRelations>
</closure>
```

Caution: For good performance it is recommended specifying the traversable artifacts/relations so that parts of the repository which will not produce any results will not be searched. The report builds quickly when `query` attributes are not used frequently. In debug mode you can check the report generation times and trace the search for artifact closure.

Chapter 13: System Administration

The administrator may need to perform additional administration tasks that are not accessible from the user interface.

This chapter includes details of tasks are performed outside the user interface.

- "Using the Audit Log" below

  Use the audit log to monitor Systinet.

- "Data Migration" on page 212

  Migrate data from previous versions of HP Systinet to Systinet.

- "Configuration Procedures" on page 215

  Manage the configuration of Systinet using the command-line utilities.

Using the Audit Log

The audit log enables you to track all the significant events and user actions in Systinet.

The log is available in `systinet_audit.log` in the application server log folder for your deployment.

Each line in the log contains the following comma separated content:

Audit Log Content

Content	Description
Time	The date and time of the event.
Category	The category displays a lowercase prefix representing the application component related to the event and an uppercase suffix representing the particular event.
Event ID	A unique code for the event.
Result	The result of the event, usually `INFO` or `ERROR`.
Context ID	The ID of the process related to the event.
Actor	The name of the user identity who performed the action.
Resource ID	The ID of the primary entity or entities involved in the event.
Detail	Readable text with a description of the event.

The entries in the log are described in the following tables organized by the category prefix representing the component associated with the event:

- "application Events"
- "application.license.check Events"
- "application.license.check.concurrent-users Events"
- "application.license.management.user-seat Events"
- "application.notification Events"
- "application.notification.email Events"
- "application.security.account Events"
- "application.security.acl Events"
- "application.security.ownership Events"
- "integration.discovery Events"
- "integration.discovery.mark Events"
- "integration.export Events"
- "integration.synchronization Events"
- "lifecycle.approval Events"
- "lifecycle.artifact Events"
- "lifecycle.automatic-action Events"
- "lifecycle.process Events"
- "platform.contract-management Events"
- "platform.repository.artifact Events"
- "platform.repository.bulk Events"
- "platform.repository.domain Events"
- "platform.repository.exportimport Events"
- "platform.repository.legacy.artifact Events"
- "platform.repository.legacy.collection Events"
- "platform.rest.atom Events"

- "platform.rest.legacy Events"
- "policy-manager.validation Events"
- "webui.authorization Events"
- "webui.document Events"
- "webui.request Events"

application Events

Category Suffix	Event ID	Resource	Result	Description
STARTUP	10000	Servlet Context	INFO	Application started.
SHUTDOWN	10001	Servlet Context	INFO	Application shut down.

application.license.check Events

Category Suffix	Event ID	Resource	Result	Description
VERIFICATION_ FAILED	10100		ERROR	Invalid license key. Systinet is unavailable until you enter a valid license key using the Setup Tool.
SYSTEM_ DATE_ CHANGED	10101		ERROR	System date change. Systinet is unavailable until you adjust the system date to be consistent with the installation time.
PRODUCT_ MISMATCH	10102		ERROR	License key for a different product. Systinet is unavailable until you enter a valid license key using the Setup Tool.
VERSION_ MISMATCH	10103		ERROR	License key for a different version. Systinet is unavailable until you enter a valid license key using the Setup Tool.
EXPIRATION	10120		WARNING / ERROR	License is due to expire or expired on the specified date. Visit the HP Software Support (SSO) web site (you must have an HP Portal account) or contact your sales representative to obtain a new license.

application.license.check Events, continued

Category Suffix	Event ID	Resource	Result	Description
NOT_LICENSED	10121		INFO	Policy Manager is not enabled in the current license.

application.license.check.concurrent-users Events

Category Suffix	Event ID	Resource	Result	Description
EXCEEDED	10160		ERROR	The number of concurrent users exceeds the current license limit. Visit the HP Software Support (SSO) web site (you must have an HP Portal account) or contact your sales representative to obtain a new license.
PERCENT_95	10161		WARNING	The number of concurrent users is approaching the license limit.Visit the HP Software Support (SSO) web site (you must have an HP Portal account) or contact your sales representative to obtain a new license.

application.license.management.user-seat Events

Category Suffix	Event ID	Resource	Result	Description
ASSIGNED	10140		INFO	License seat assigned to the specified user.
REMOVED	10141		INFO	License seat removed from the specified user.

application.notification Events

Category Suffix	Event ID	Resource	Result	Description
FAILED	10230		ERROR	Unsuccessful notification.

application.notification.email Events

Category Suffix	Event ID	Resource	Result	Description
FAILED	10200		ERROR	E-mail delivery failure with failure details.
ENQUEUED	10201		INFO	E-mail queued for delivery.
SENT	10202		INFO	E-mail delivery success.

application.security.account Events

Category Suffix	Event ID	Resource	Result	Description
CREATE	10300	Username	INFO / ERROR	Specified user account created or creation failed.
DELETE	10301	Username	INFO / ERROR	Specified user account deleted or deletion failed.

application.security.acl Events

Category Suffix	Event ID	Resource	Result	Description
CHANGE	10400	Resource ID	INFO / ERROR	ACL of specified resources changed or change failed.

application.security.ownership Events

Category Suffix	Event ID	Resource	Result	Description
RETIREMENT	10900	Retired Owner	INFO	User retired and owned artifacts transferred to the specified user or group.

integration.discovery Events

Category Suffix	Event ID	Resource	Result	Description
START	11000	Server UUID	INFO	BSM Discovery started for the specified server.
END	11001	Server UUID	INFO	BSM Discovery complete for the specified server with a link to the report.
FAILED	11002	Server UUID	ERROR	BSM Discovery failed for the specified server with a link to the report.

integration.discovery Events, continued

Category Suffix	Event ID	Resource	Result	Description
ARTIFACT	11003	Artifact UUID	INFO / ERROR	The specified artifact was discovered or an error during discovery.

integration.discovery.mark Events

Category Suffix	Event ID	Resource	Result	Description
GOVERNED	11031	Artifact UUID	INFO / ERROR	Specified artifact entered into governance or an error.
INFRASTRUCTURE	11032	Artifact UUID	INFO / ERROR	Specified artifact marked as infrastructure or an error.
ROGUE	11005	Artifact UUID	INFO / ERROR	Specified artifact marked as rogue or an error.

integration.export Events

Category Suffix	Event ID	Resource	Result	Description
START	11200	Server Artifact Name	INFO	Export to the specified server started.
END	11201	Server Artifact Name	INFO	Export to the specified server complete.
FAILED	11202	Server Artifact Name	ERROR	Export to the specified server failed.
ARTIFACT	11203	Artifact UUID	INFO / ERROR	Specified artifact exported or export failed.

integration.synchronization Events

Category Suffix	Event ID	Resource	Result	Description
UPDATE	11100	Artifact UUID	INFO / ERROR	Specified artifact updated or update failed during synchronization.

lifecycle.approval Events

Category Suffix	Event ID	Resource	Result	Description
PROMOTION_ START	15700	Artifact UUID	INFO / ERROR	Promotion for the specified artifact started or failed to start.
PROMOTION_ END	15701	Artifact UUID	INFO	Promotion complete for the specified artifact.
PROMOTION_ FAILED	15702	Artifact UUID	ERROR	Promotion failed for the specified artifact.
PROMOTION_ CANCELLED	15703	Artifact UUID	INFO	Promotion cancelled for the specified artifact.
REQUEST_ APPROVED	15704	Approval UUID	INFO / ERROR	Promotion request approval or request failure.
REQUEST_ DENIED	15705	Approval UUID	INFO / ERROR	Promotion request denial or denial failure.

lifecycle.artifact Events

Category Suffix	Event ID	Resource	Result	Description
GOVERN	15000	Artifact UUID	INFO / ERROR	Starting governance for the specified artifact using the specified lifecycle process or a governance failure.
UNGOVERN	15001	Artifact UUID	INFO / ERROR	Ending governance for the specified artifact or end governance failure.

lifecycle.automatic-action Events

Category Suffix	Event ID	Resource	Result	Description
START	15030	Process UUID	INFO / ERROR	Specified automatic action executed or failed to execute.

lifecycle.process Events

Category Suffix	Event ID	Resource	Result	Description
CREATE	15500	Process UUID	INFO / ERROR	The specified lifecycle process created or creation failed.

lifecycle.process Events, continued

Category Suffix	Event ID	Resource	Result	Description
UPDATE	15501	Process UUID	INFO / ERROR	The specified lifecycle process updated or update failed.
DELETE	15502	Process UUID	INFO / ERROR	The specified lifecycle process deleted or deletion failed.

platform.contract-management Events

Category Suffix	Event ID	Resource	Result	Description
REQUEST_ CREATE	14000	Resource Path	INFO / ERROR_ ACCESS_ DENIED	Contract request creation or failed due to insufficient permissions.
REQUEST_ DELETE	14001	Resource Path	INFO	Contract request deletion.
REQUEST_ ACCEPT	14002	Resource Path	INFO / ERROR / ERROR_ ACCESS_ DENIED	Contract request accepted or failed or failed due to insufficient permissions.
REQUEST_ REJECT	14003	Resource Path	INFO / ERROR_ ACCESS_ DENIED	Contract request rejection or failed due to insufficient permissions.
CONTRACT_ CREATE	14004	Resource Path	ERROR_ ACCESS_ DENIED	Contract creation failure due to insufficient permissions.
CONTRACT_ REVOKE	14005	Resource Path	INFO / ERROR / ERROR_ ACCESS_ DENIED	Contract revoked or failed or failed due to insufficient permissions.
CONTRACT_ INVALIDATE	14006	Resource Path	INFO / ERROR	Contract invalidation or invalidation failure.

platform.repository.artifact Events

Category Suffix	Event ID	Resource	Result	Description
CREATE	12000	Artifact UUID	INFO / ERROR	Artifact creation or creation error.
UPDATE	12001	Artifact UUID	INFO / ERROR	Artifact update or update error.
DELETE	12002	Artifact UUID	INFO / ERROR	Artifact deletion or deletion error.
UNDELETE	12003	Artifact UUID	INFO / ERROR	Artifact undeletion or undeletion error.
PURGE	12004	Artifact UUID	INFO / ERROR	Artifact purging or purge error.

platform.repository.bulk Events

Category Suffix	Event ID	Resource	Result	Description
CHANGE_ OWNER	12030	Artifact UUIDs	INFO / ERROR / ERROR_ PARTIAL / ERROR_ ACCESS_ DENIED	Artifact change owner to the specified owner for the specified artifacts or errors.

platform.repository.domain Events

Category Suffix	Event ID	Resource	Result	Description
CREATE	19500	Domain ID	INFO	The specified domain created.
UPDATE	19501	Domain ID	INFO	The specified domain updated.
DELETE	19502	Domain ID	INFO	The specified domain deleted.

platform.repository.exportimport Events

Category Suffix	Event ID	Resource	Result	Description
EXPORT_ START	19200	image name	INFO	Repository export started.
EXPORT_END	19201	image name	INFO	Repository export finished.

platform.repository.exportimport Events, continued

Category Suffix	Event ID	Resource	Result	Description
IMPORT_START	19202	image name	INFO	Repository import started.
IMPORT_END	19203	image name	INFO	Repository import finished.
UNDO_START	19204	image name	INFO	Repository import rollback started.
UNDO_END	19205	image name	INFO	Repository import rollback finished.

platform.repository.legacy.artifact Events

Category Suffix	Event ID	Resource	Result	Description
CREATE	12200	Artifact Path	INFO / ERROR	Artifact creation or creation error.
UPDATE	12201	Artifact Path	INFO / ERROR	Artifact update or update error.
DELETE	12202	Artifact Path	INFO / ERROR	Artifact deletion or deletion error.
REMOVE	12203	Artifact Path	INFO / ERROR	Artifact removal or removal error.
UNDELETE	12204	Artifact Path	INFO / ERROR	Artifact undeletion or undeletion error.
PURGE	12205	Artifact Path	INFO / ERROR	Artifact purging or purge error.
SECURITY_ PROPERTIES	12206	Artifact Path	INFO / ERROR	Artifact security modification or modification error.

platform.repository.legacy.collection Events

Category Suffix	Event ID	Resource	Result	Description
CREATE	12230	Artifact Path	INFO / ERROR	Artifact collection creation or creation error.
DELETE	12231	Artifact Path	INFO / ERROR	Artifact collection deletion or deletion error.
PURGE	12232	Artifact Path	INFO / ERROR	Artifact collection purging or purge error.

platform.repository.legacy.collection Events, continued

Category Suffix	Event ID	Resource	Result	Description
UPDATE	12233	Artifact Path	INFO / ERROR	Artifact collection modification or modification error.
SECURITY_ PROPERTIES	12234	Artifact Path	INFO / ERROR	Artifact collection security modification or modification error.

platform.rest.atom Events

Category Suffix	Event ID	Resource	Result	Description
GET	12700	Request Path	INFO	HTTP GET method invoked by the specified URI.
HEAD	12701	Request Path	INFO	HTTP HEAD method invoked by the specified URI.
POST	12702	Request Path	INFO	HTTP POST method invoked by the specified URI.
PUT	12703	Request Path	INFO	HTTP PUT method invoked by the specified URI.
DELETE	12704	Request Path	INFO	HTTP DELETE method invoked by the specified URI.

platform.rest.legacy Events

Category Suffix	Event ID	Resource	Result	Description
GET	12500	Request URI	INFO	HTTP GET method invoked.
POST	12501	Request URI	INFO	HTTP POST method invoked.
PUT	12502	Request URI	INFO	HTTP PUT method invoked.
DELETE	12503	Request URI	INFO	HTTP DELETE method invoked.
HEAD	12504	Request URI	INFO	HTTP HEAD method invoked.

policy-manager.validation Events

Category Suffix	Event ID	Resource	Result	Description
DELETE	13000		INFO / ERROR	Validation deleted or failed to delete.

policy-manager.validation Events, continued

Category Suffix	Event ID	Resource	Result	Description
START	13001	Validation Requests	INFO / ERROR	Validation started or failed to start.
START_ PERIODIC	13002		INFO / ERROR	Periodic validation started or failed to start.
CANCEL	13003	Execution ID	INFO / ERROR	Cancelling a validation succeeded or failed.
START_JMS	13004		INFO / ERROR	Starting JMS validation succeeded or failed.
END_JMS	13005		INFO / ERROR	Ending JMS validation succeeded or failed.

webui.authorization Events

Category Suffix	Event ID	Resource	Result	Description
LOGIN	10600		INFO / ERROR	User sign-in or sign-in failed.
LOGOUT	10601		INFO	User sign-out.

webui.document Events

Category Suffix	Event ID	Resource	Result	Description
ACCESS	16100		INFO	The specified artifact accessed.
EDIT	16101		INFO	The specified artifact edited.

webui.request Events

Category Suffix	Event ID	Resource	Result	Description
GET	17200	Relative URL	INFO	HTTP GET method invoked at the relative URL.
HEAD	17201	Relative URL	INFO	HTTP HEAD method invoked at the relative URL.
POST	17202	Relative URL	INFO	HTTP POST method invoked at the relative URL.
PUT	17203	Relative URL	INFO	HTTP PUT method invoked at the relative URL.

webui.request Events, continued

Category Suffix	Event ID	Resource	Result	Description
DELETE	17204	Relative URL	INFO	HTTP DELETE method invoked at the relative URL.

Configuring the Audit Log for JBoss

In JBoss it is possible to restrict the categories that Systinet writes to the audit log and also configure the event details.

To configure the audit log categories in JBoss

1. Open `JBOSS_HOME/server/CONFIG_HOME/conf/jboss-log4j.xml` with a text editor.

   The default configuration is the following:

   ```
   <!-- ============================== -->
   <!-- HP Systinet Audit category -->
   <!-- ============================== -->
   <category name="com.hp.systinet.audit.event" additivity="true">
     <priority value="INFO"/>
     <appender-ref ref="Systinet_AUDIT"/>
   </category>
   ```

2. Make the logging more specific by changing the category.

   For example for `lifecycle.api` category events:

   ```
   <category name="com.hp.systinet.audit.event.lifecycle.api" additivity="true">
     <priority value="INFO"/>
     <appender-ref ref="Systinet_AUDIT"/>
   </category>
   ```

3. Add a separate category section for each specific category that you want to log.

4. Save `JBOSS_HOME/server/CONFIG_HOME/conf/jboss-log4j.xml`.

To configure the audit event content in JBoss

1. Open `JBOSS_HOME/server/CONFIG_HOME/conf/jboss-log4j.xml` with a text editor.

   The default configuration is the following:

   ```
   <!-- ========================= -->
   <!-- HP Systinet Audit Appender -->
   ```

```
<!-- ======================== -->
<appender class="org.apache.log4j.RollingFileAppender"
    name="Systinet_AUDIT">
  <param name="File" value="${jboss.server.log.dir}/systinet_audit.log"/>
  <param value="10000KB" name="MaxFileSize"/>
  <param value="10" name="MaxBackupIndex"/>
  <layout class="org.apache.log4j.PatternLayout">
    <!--" is the same as " (")-->
    <param name="ConversionPattern"
        value=""%d",%X{audit.category}:%X{audit.eventId},
        %X{audit.result},%X{audit.ctxId},"%X{audit.actor}",
        "%X{audit.resource}",%X{audit.detail}%n"/>
  </layout>
</appender>
```

2. Rearrange, remove, and add to the audit event layout conversion pattern as required.

   Note: The default configuration contains all the available audit content.

3. Save `JBOSS_HOME/server/CONFIG_HOME/conf/jboss-log4j.xml`

Data Migration

Systinet 10.01 is not backwards compatible with Systinet 10.00 and 4.x data. You can import data images from Systinet 10.00 and 4.x into Systinet 10.01 using a migration tool provided in the installation.

Note: Systinet 10.01 migrate tool supports Systinet 10.00, 4.03 and 4.10 only. If you are migrating from a version of Systinet earlier than 4.03 or if you encounter problems during custom migration, contact HP Professional Services for assistance.

Tip: Prior to migration, HP recommends purging activity reports and recreating the Activity Report Task. There may be thousands of these reports or revisions of them due to internal reporting activity and removing them may significantly reduce the migration process time.

To Remove Activity Reports in Systinet 10.00 and 4.x:

1. Open **View Reports -> All** in the Tools tab.

2. Filter the reports, using name `Activity Report`.

3. Use the selection drop-down and **Select All**.

4. Expand **Select Action**, and select **Delete**.

5. Select **Non-Recoverable Deletion** and **Ignore Incoming Artifacts**, and leave **Delete Sub-Artifacts** unselected.

6. Confirm the deletion.

   > **Note:** The deletion may take some time.

7. Open the detail view of the **Activity Report Update Task** in the Tools tab.

8. Delete the task with Non-Recoverable Deletion option selected.

9. Create a new task using the following parameters:

Parameter	Value
Name	Activity Report Update Task
Tool	Activity Report update job
Recurrence	Daily

To Migrate Data from Systinet 10.00 and 4.x to 10.01:

1. In Systinet 10.00 and 4.x, execute the export command:

   SYSTINET_HOME/bin/export -image *dataimage.zip*

   For details, see the "Administration Utilities" section.

2. In Systinet 10.01, execute the data migration command:

   SYSTINET_HOME/bin/migrate --image *dataimage.zip* --output *migratedimage.zip*

   > **Note:** Execute **migrate --help** to view the available options for the migrate tool. If you use password encryption, use the passphrase setup for Systinet 10.01 if it is different from that of Systinet 10.00 and 4.x.

   The migrate tool creates an image folder matching the output of the export tool ready for import to Systinet 10.01. The validate switch performs XML schema validation on the resulting data image. If errors occur, it typically indicates that your deployment has some non-standard customization. Depending on the type of error, you need to either follow the upgrade process described in "Apply Custom Extensions from HP SOA Systinet 3.x" on page 1 or contact HP Technical Support. The migration tool logs progress to `SYSTINET_HOME/log/migrate.log`. When some error occurs during the migration, it is logged in this file.

3. In Systinet 10.01, execute the import command line:

   SYSTINET_HOME/bin/import --image *migratedimage.zip*

   For more details, see "Import Tool" on page 256

   > **Caution:** The import should not be run using the **--force** switch. This can overwrite built-in core data, such as taxonomies, with data from 10.00 and 4.x which may impact server functionality. Only use **--force** if you know exactly what the effect is.

Details of the migration are reported to a log file accessible at `SYSTINET_HOME/log/migrate.log`.

> **Note:** HP recommends updating Oracle Database schema statistics after importing large amounts of data. Old statistics may impact the performance of some data queries. Consult your database administrator.

To Update Oracle Schema Statistics:

- Execute the following command:

  EXEC DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '&1',no_invalidate => FALSE,options => 'GATHER');

  This command does not require database admin privileges and can be run by the schema owner (ownname).

Pay particular attention to the following migrations:

- **Model Change**
  SDM is extensively changed from 10.00 and 4.x to 10.01. For details, see “Model Changes” in the *Reference Guide*.

- **Lifecycle-Based Contracts**
  Contract states are extensively re-designed to employ the standard Lifecycle feature. For details, see “Contract Management” in the *User Guide*.

- **Group Membership**

  During import, the group membership of the migrated image is merged with any existing group membership.

  > **Note:** Import of a 10.00 and 4.x image replaces the current group membership with the imported group membership if they exist.

- **UI Customizations**

UI customizations from Systinet 10.00 and 4.x migrate are migrated along with data image. Use the UI customization features in the Administration tab in Systinet 10.01 to verify and change these customizations after data migration is complete. For details, see "UI Customization" on page 109.

The following data from Systinet 10.00 and 4.x are not migrated to 10.01:

- **Rebranding** - Rebranding is part of server installation. To rebrand the new Systinet 10.01 installation, see "Rebranding Systinet".

Configuration Procedures

In addition to the Configuration Management provided in the user interface, system configuration is maintained using the command-line tools described in "Administration Utilities" on page 252. These tools enable you to reset the parameters set during installation. They also enable you to make additional changes to your installation.

This section describes the processes executed by the provided tools that are not fully described as part of the installation process:

- "Applying Extensions" below
- "Applying Updates" on page 219
- "Setting Up LDAP Integration" on page 221
- "Setting Up SiteMinder Integration" on page 221
- "SSL Certificates" on page 222
- "Redeploying the EAR File" on page 227
- "Disabling Default Domain Sign-In" on page 228
- "Rebranding Systinet" on page 228
- "Configuring Versioning" on page 231

For details about the configuration that the administrator can perform using the user interface, see "Configuration Management" on page 76.

Applying Extensions

You can extend 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 Systinet come from the following sources:

- **Customization Editor**

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

  > **Caution:** If your extension contains new artifact types, Systinet does not create default ACLs for them. Set default ACLs for the new artifact types in Systinet using the functionality described in "How to Manage Default Access Rights" on page 20.

- **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 the next page

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

- "Decoupled DB Scenario" on page 218

  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.

> **Caution:** In some specific circumstances (underscores and numbers in property names), extension application may fail because 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 Systinet using an older table naming algorithm in order to preserve backward compatibility with HP 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 `SYSTINET_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 Systinet.

To apply extensions to Systinet in a single step:

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

   ```
   SYSTINET_HOME/extensions
   ```

   The Setup Tool automatically applies all extensions in that directory.

   > **Note:** If you are applying extensions to another server, substitute the relevant home directory for `SYSTINET_HOME`.

2. Stop the server.

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

   SYSTINET_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.

   > **Note:** 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 227.

8. Restart the server.

> **Caution:** Applying an extension that modifies the SDM model may drop your full text indices. `SYSTINET_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 following sections of the *Installation and Deployment Guide*:

- "Enable Full-Text Search in DB2"
- "Enable Full-Text Search in MSSQL"
- "Enable Full-Text Search in Oracle"

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:

   ```
   SYSTINET_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:

   SYSTINET_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 `SYSTINET_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:

   SYSTINET_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 227.

Applying Updates

HP Software provide updates to Systinet between major releases. HP Software Support notify you of any updates recommended for your deployment, but it is also possible to obtain them from the HP website.

To locate updates:

1. Open the HP website: http://www.hp.com

2. In the Customer Support section, click **Software & Driver Downloads**.

3. Select the **Software Products→IT Management Software** link from the product categories.

4. In the Where Do I Find Section, select **Software Patches**.

   Note: This page requires an HP Passport. Login or register for a passport with a username and a password containing at least 6 characters including a number and a non-alphanumeric character.

   The Patches Search page opens.

5. Select the following search terms:

Parameter	Search Term
Product	HP Systinet

Parameter	Search Term
Version	As appropriate.
Sub-Product	HP SystinetPlatform

6. Click **Search** to see the list of updates for the selected version of the product.

7. Select an update name to see the details of that particular update including installation instructions.

8. Select **DOWNLOAD PATCH** to download the update.

9. Install the update with reference to the specific installation instructions and the procedure "To apply updates:" below.

Product updates from HP Software are applied to Systinet using the Setup Tool. For more details see "Setup Tool" on page 249.

To apply updates:

1. Copy the updates to the `SYSTINET_HOME/updates` folder.

2. Stop the application server.

3. Start the Setup Tool by executing **SYSTINET_HOME/bin/setup.bat(sh)**.

4. Choose the **Apply Updates** scenario, and click **Next**.

5. Use **Add** to select the updates to apply, and then click **Next**.

   > **Note:** If you applied Step 1, this is not required.

   The Setup Tool validates the updates.

6. Click **Next** to apply the updates.

7. Click **Next**, and then click **Finish** to close the Setup Tool or **Setup Again** to restart the Setup Tool.

8. Reapply any required extensions, using the procedure described in "Applying Extensions" on page 215.

9. Redeploy the EAR file as described in:

   - **JBoss** - Follow the procedure described in "Redeploying the EAR File" on page 227.

10. Restart the application server.

    See also "Repository Data Diagnostic Tool" on page 108

Setting Up LDAP Integration

The Setup Tool enables you to integrate LDAP accounts and groups after installation.

To Integrate LDAP using the Setup Tool:

1. Execute **SYSTINET_HOME/bin/setup**, and click **Next**.
2. In the Select Scenarios page, select **Advanced**, and click **Next**.
3. In the Custom Scenario Selection page, select **User Management**, and click **Next**.
4. In the Account Provider Selection page, select **LDAP**, and click **Next**.
5. The Setup Tool continues with the LDAP setup pages identical to the GUI Installation wizard.

   For details, see "LDAP Options" in the *Installation and Deployment Guide*.
6. Do one of the following:
   - Click **Finish** to exit the Setup Tool.
   - Click **Setup Again** to perform another task using the Setup Tool.

Note: Systinet logins are case-insensitive by default. If want the login name to be case-sensitive you must set the `shared.um.account.caseInsensitiveLoginName` property to `false`. For details, see "How to Manage System Settings" on page 81. You must also ensure that the application server uses matching case-sensitive or -insensitive authentication as well.

Setting Up SiteMinder Integration

You can configure Systinet to accept authentication headers or cookies added to HTTP requests after a successful authentication performed by an authentication proxy. The changes affect the configuration properties stored in the database and the application EAR file.

To Integrate Siteminder using the Setup Tool:

1. Execute **SYSTINET_HOME/bin/setup**, and click **Next**.
2. In the Select Scenarios page, select **Advanced**, and click **Next**.
3. In the Custom Scenario Selection page, select **Siteminder Setup**, and click **Next**.

4. In the Siteminder Setup page, select **Enable Siteminder Integration** and then click **Next**.

   Note: If you are disabling Siteminder integration, deselect **Enable Siteminder Integration** and click **Next**. Continue with **Step 8**.

5. Do one of the following:

   - Select **Use Cookies** to accept authentication cookies.

   - Select **Use Headers** if the user login name is sent in the authentication header.

6. Set the Login Header or Cookie Name and then click **Next**.

7. After deployment validation, click **Next** to start the setup.

   The Setup Tool updates your deployment and configuration.

8. After setup completes, click **Next** and do one of the following:

   - Click **Finish** to exit the Setup Tool

   - Click **Setup Again** to perform another task using the Setup Tool.

9. Redeploy the EAR file as described in:

   - **JBoss**

     Follow the procedure described in "Redeploying the EAR File" on page 227.

10. Restart the application server.

SSL Certificates

Systinet can use HTTPS for communicating with other applications, such as UDDI registries, and *HP Business Service Management*, or to publish resources, such as WSDL and XML schemas, accessible via HTTPS.

HTTPS runs on top of SSL and requires additional set up steps. SSL requirements are driven by the target server, which can operate in one of the following modes:

1. **One-way SSL**

   Mostly used for WWW, where the server authenticates to the client by providing its certificate.

   The following conditions must be met:

   - The server certificate must be valid and trusted.

   - The resource HTTPS URL must contain a host name contained in the target server SSL certificate.

2. **Required two-way SSL**

   Mutual authentication is required and both server and client authenticate each other using certificates.

   The following conditions must be met:

   - The requirements for one-way SSL.
   - The client must provide its certificate, and the server must trust the client certificate.

3. **Wanted two-way SSL**

   Mutual authentication is wanted but not mandatory and the connection can be established as one-way SSL.

   The requirements are the same as for one-way SSL.

Using SSL is described in the following sections:

- "Identifying Server SSL Requirements" below
- "SSL Server Certificate Trust" on the next page
- "Importing Client Certificates for Two-Way SSL" on the next page
- "SSL Customization" on page 225
- "SSL Troubleshooting" on page 227

Identifying Server SSL Requirements

Systinet deployed to an application server acts as an HTTPS client.

The SSL requirements of the target server can be determined using the ssltool provided with Systinet.

For example, with the command:

ssltool serverInfo --url https://localhost:8443 --certFile localhost.crt

The output resembles the following:

```
localhost:2968
Host: localhost/127.0.0.1
Port: 2968
Client authentication: WANTED
Accepted issuers of client certificates:
 OU=Some CA, CN=CA
Server Certificate:
 Subject: CN=localhost
 Issuer: OU=Some CA, CN=CA
 Issued for hostname(s): localhost
 Algorithm: RSA
 Valid from: 12/7/07 4:22 PM
 Valid to: 12/4/17 4:22 PM
Received Server Certificate Chain:
 CN=localhost
 OU=Some CA, CN=CA
PEM Encoded Received Server Certificate Chain:
-----BEGIN CERTIFICATE-----
  ...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
  ...
-----END CERTIFICATE-----
```

The output contains the following elements:

- The SSL server mode is output after `Client Authentication`, where:
  - `NOT_REQUESTED` means one-way SSL.
  - `WANTED` means non-mandatory two-way SSL.
  - `REQUIRED` means required two-way SSL.
- The accepted issuers for two-way SSL. These display the subject names of accepted certificate authorities that can issue client certificates.
- The server certificate details saved to a file, `localhost.crt`.
- Host names that must be used in URLs are listed after `Issued for hostname(s):`.
- PEM encoded certificates that must be trusted.

For more details about the SSL Tool, see "SSL Tool" on page 267.

SSL Server Certificate Trust

By default, any SSL server certificate is trusted.

Note: For non-default options, see "SSL Customization" on the next page .

Importing Client Certificates for Two-Way SSL

A client certificate (a private key with an associated certificate) issued by a certificate authority trusted by the server must be imported to the keystore. By default, this keystore is the one used by JDK.

Usually it is empty or unspecified. The location of the JDK SSL keystore is determined using a procedure explained at http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html#X509KeyManager. JDK keystores are managed by the keytool.

For details about keytool, see http://java.sun.com/j2se/1.5.0/docs/tooldocs/index.html#security.

Note: For non-default options, see "SSL Customization" below.

SSL Customization

Systinet customizes SSL trust, key management, and hostname verification. There are four available SSL customizations available after installation. The default customization is named `skipped` and this is the customization used in the previous procedures in this section.

The list of available customizations is available with the command:

ssltool customize --info

The output should resemble the following:

```
Effective customization: composite
Available customizations:
--> default
  Java/JSSE default key/trust stores, default hostname verifier.
--> skipped
  Server certificates are always trusted and pass hostname verification, default
  key manager is used.
--> database
  Database key/trust stores, default hostname verifier.
--> composite
  Composition of database and default key/trust stores, default hostname verifier.
```

- default

  This means no customization and the application server trust and keystore is used.

  Hostname verification is required.

- skipped

  This means that server certificates are ignored as no trust is required.

  Hostname verification is ignored.

  Client certificates are taken from a file described here at http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html#X509KeyManager.

  The `skipped` customization is a default, it is chosen during installation by deselecting **Verify Certificates**.

- database

  This means that only the Systinet database is used for the trust and keystore.

  Hostname verification is required.

- composite

  This means that a composite of the JDK truststore and Systinet database truststore is used.

  A certificate is trusted if it is in either the JDK or the Systinetdatabase truststore.

  Client certificates are taken from both the JDK and Systinet database keystores.

  Hostname verification is required.

  The `composite` customization can be chosen during installation by selecting **Verify Certificates**.

You can manage the database keystore and truststore using the ssltool provided with Systinet. These stores are specific to Systinet. Unlike the JDK keystore, changing the database keystore does not require a restart of the application server, because changes take effect in approximately thirty seconds.

To Import a Server Certificate to the Database Truststore:

- Execute the following command:

  ```
  SYSTINET_HOME/bin/ssltool keystoreEl -i truststore --certfile CERTIFICATE_FILE --add
  ```

To import a Client Certificate to the Database Keystore:

- Execute the following command:

  ```
  SYSTINET_HOME/bin/ssltool keystoreEl -i keystore --keystore CLIENT_CERT --storepass PASSWORD --add
  ```

  CLIENT_CERT can be either a Java keystore with a private key or a p12 file (PKCS 12). *PASSWORD* is the password used to protect the file.

For more details about the SSL Tool, see "SSL Tool" on page 267.

To Change the SSL Customization:

- Do one of the following:
  - Change the setting in the Systinet Configuration page.

    For details, see "How to Manage Basic Configuration Options" on page 77.

  - Execute the following command:

    ```
    SYSTINET_HOME/bin/ssltool customize --change CUSTOMIZATION_NAME
    ```

SSL Troubleshooting

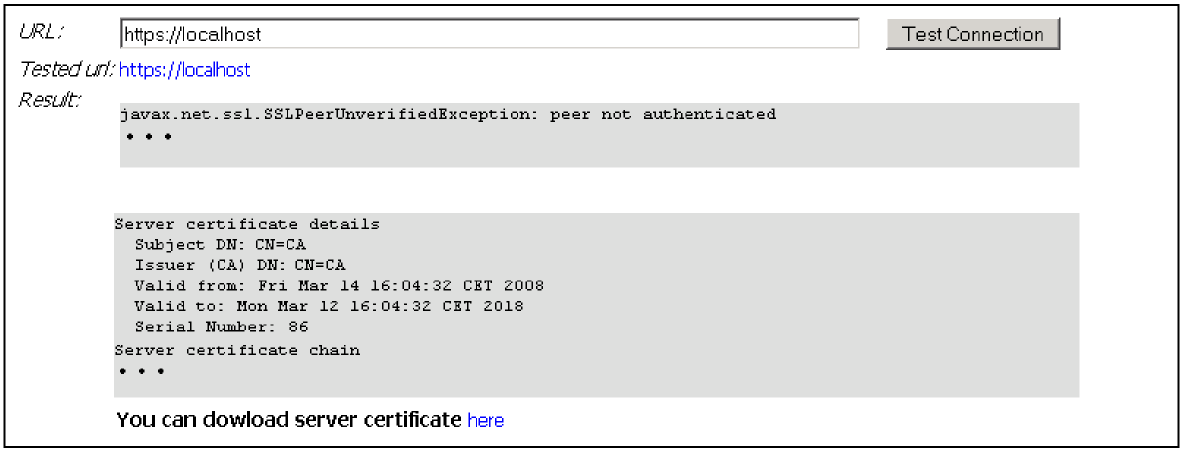
You can test the SSL set up for a particular HTTPS URL using the self-tester.

The test page is available at the following URL:

```
http://localhost:port/context/self-test/self-http-test
```

Type the URL to check and click **Test Connection**.

The output resembles the following:



```
javax.net.ssl.SSLPeerUnverifiedException: peer not authenticated
 . . .
```

```
Server certificate details
  Subject DN: CN=CA
  Issuer (CA) DN: CN=CA
  Valid from: Fri Mar 14 16:04:32 CET 2008
  Valid to: Mon Mar 12 16:04:32 CET 2018
  Serial Number: 86
Server certificate chain
 . . .
```

You can dowload server certificate here

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.

To Redeploy the EAR file to JBoss:

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

   SYSTINET_HOME/bin/setup.bat(sh)

3. Select the **Advanced** scenario, and click **Next**.
4. Scroll down, select **Deployment**, and then click **Next**.

   Once 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.

Disabling Default Domain Sign-In

When Systinet installs there is a default domain which users sign-in to if they are not assigned to specific working domains.

The Administration UI enables you to change which domain is the default domain but in certain circumstances (for example, where you have thousands of LDAP users) you may want to disable default domain functionality so that users cannot sign-in unless they have been specifically assigned to a domain.

Disabling the default domain can only be done by manually modifying the configuration properties file.

To Disable Default Domain Sign-In:

1. Stop the application server.

2. Export the HP Systinet configuration with the following command:

   SYSTINET_HOME/bin/export --image *IMAGE* --components configuration

3. Open `IMAGE/configuration/configuration-properties.xml` in a text editor.

4. Change the property `platform.catalog.defaultUiAdapter.defaultDomain`, setting the value to an empty string.

5. Save `configuration-properties.xml`.

6. Upload the new configuration with the following command:

   SYSTINET_HOME/bin/import --image *IMAGE* --components configuration --configuration-application --configuration-system

7. Restart the application server.

Note: When default domain sign-in is disabled, the administrator must enable users by creating user artifacts or importing them from the user store, and then assigning them to a specific domain.

Rebranding Systinet

Systinet enables you to rebrand the product to match your corporate image.

To rebrand Systinet

1. Stop the Systinet server.

2. Extract the default image with the following command:

SYSTINET_HOME/bin/rebrand --extract --image *IMAGE*

IMAGE is the path to the directory to contain the extracted extracted brand image.

3. Navigate to the brand image and extract the `rebranding-image-TIMESTAMP.zip` file.

4. Modify the brand content as required. For details of the brand image content see "Brand Image Content" on the next page

5. Apply the image to Systinet with the following command:

   SYSTINET_HOME/bin/rebrand --apply --image *IMAGE*

   Note: You can archive the image folder and execute the following command:

   SYSTINET_HOME/bin/rebrand --apply --image *IMAGE.zip*

6. Redeploy the EAR file as described in:

   - **JBoss**

     Follow the procedure described in "Redeploying the EAR File" on page 227.

   HP recommends precompiling JSPs after the **rebrand --apply** step and before deploying the EAR file. Any precompilation applied prior to the **rebrand** command is lost.

7. Restart the Systinet server.

8. The default administrator name is branded as *HP Systinet Administrator*. If you want to change the administrator name after installation, login as the administrator and edit the profile name as described in "How to Manage Your Profile" in the *User Guide*.

9. The product documentation is HP branded. To change the documentation branding accessed from **Help** in all product UI pages, in addition to replacing the images in the `doc` folder in the brand image, edit the `<title>` element in `SYSTINET_HOME/doc/WebHelp/Data/Skin_HP_WebHelpSkin_no_Index/Skin.xml`. Make sure your changes also apply in the hp-systinet-doc.war in SYSTINET_HOME/deploy directory before you use the setup tool to deploy files to your application server.

   Note: The product name is used throughout the text of both the HTML and PDF documentation. These references cannot be rebranded.

10. Many Systinet reports include an HP branded logo in the header. For details of rebranding this image, see "Rebranding Systinet Reports" in the *Report Editor Guide*.

    Note: Instead of exporting all reports from Report Editor, you can add the modified reporting library to the rebranding image before import.

For more command line options for the rebrand tool, see "Rebrand Tool" on page 265.

Tip: The first time you execute **rebrand --extract**, Systinet creates a default image archive, `default-image.zip`, in `SYSTINET_HOME/conf/rebranding`. Use this image with **rebrand --apply** to restore Systinet to the default image.

Brand Image Content

After extracting a brand image archive the image consists of the following content:

Brand Image Content

Content	Description
`image.xml`	A file containing a description of the image files and a mapping between their location in the brand image and their location in the product EAR file described in "Brand Images" below. **Note**: If you rename image content, you must change the `imageLocation` attribute for the relevant file element. Do not change `productLocation` attributes.
`bundles`	A directory containing the resource bundle `messages.properties`. In this file you can modify the company name and add a copyright notice. For details, see "Brand Properties in rebranding.properties" below.
`data`	A directory containing default reporting library used by Report Editor.
`images`	A directory containing the branding images displayed in Systinet pages. For more details see "Brand Images" below.
`notifications`	A directory containing the Systinet mail notification header and footer.

Brand Properties in `rebranding.properties`

Property	Description
`rebranding.productName`	The product name displayed in browser titles and tabs, and throughout the user interface.
`rebranding.company`	The company name displayed throughout the user interface.
`rebranding.coCopyright.before`	An optional copyright notice placed above the HP copyright in page footers.
`rebranding.coCopyright.after`	An optional copyright notice placed below the HP copyright in page footers.
`rebranding.poweredByFooter`	An optional notice placed above the HP copyright in page footers.
`rebranding.notification.mail.subject.prefix`	The branding used in notification emails.

Brand Images

Brand Image	Product Image	Description
aboutBox_backgroundImage.png	background_aboutbox_illustration.png	The background image in the About Box. (300x300 pixels)
favicon.ico	favicon.ico	The icon displayed in the browser URL input and tab titles. Must be 16 x 16 pixels.
login_productName.png	img-login-product.png	The product name image displayed in the center of the login page. Maximum 600 x 120 pixels.
login_topRightImage.gif	img-login.jpg	The picture displayed above the credential inputs. Maximum 450 x 300 pixels.
page_headerCompanyLogo.png	product_logo.png	The company logo displayed in all pages of the UI. (200x47 pixels)
page_headerImage.png	background_header.png	The header image displayed in all pages of the UI. (550x80 pixels)
welcomeBox_navigatorImage.jpg	welcome_screen_navigator.jpg	The image in the welcome panel displayed the first time a user logs in. (250x185 pixels)
welcomeBox_topRightMenuImage.jpg	welcomescreen_illustration.jpg	The image in the welcome panel displayed the first time a user logs in. (594x76 pixels)

Configuring Versioning

Systinet provides versioning functionality that simplifies the process of publishing or creating new versions when there is data content attached to the versioned artifact. For example, a SOAP implementation is normally associated with a WSDL. When you create a new version of the SOAP service, you may also want to create a new version of the WSDL.

This section describes various versioning strategies, the default versioning configuration, and how to change the versioning strategy in the following sections:

- "Versioning Strategies" below
- "Enabling Versioning Strategies" on page 236
- "Configuring Versioning" above
- "Advanced Versioning" on page 241

Versioning Strategies

A versioning strategy describes the following functionality for published content:

- How the version number is extracted from the data content.
- How the version number determines the server folder location where Systinet stores the data content.
- How Systinet locates other versions (siblings) of an artifact.

It is important to use a consistent versioning strategy. Without one you cannot fully exploit the versioning capabilities of Systinet. For example, if a user publishes content that does not correspond to the conventions defined by the versioning strategy, then Systinet cannot establish relations to previous and next versions or they may be established incorrectly. Similarly, any manual modification to published data that does not correspond to the versioning strategy may cause problems. For details about HP best practice, see "Versioning Best Practice" in the *User Guide*.

Systinet requires versioning strategies in XML format according to the versioning schema. For details, see "Versioning Schema" in the *Reference Guide*.

The following sections describe common versioning strategies:

- "Version in Filenames" below
- "Version in Folders" on the next page
- "Version in Namespaces and Folders" on page 234
- "Version in Namespaces and Filenames" on page 235
- "Custom Versioning Strategies" on page 236

Version in Filenames

This is the default versioning behavior. Systinet expects the version number to be in the filename of the published data content.

For example, assume that you store WSDLs in a folder matching the WSDL name in the `services` folder. In this case, Systinet versions in the following way:

- The WSDL defining version 1.0 of the ATM service exists at `/services/atm/atm1.0.wsdl`
- When you create version 2.0 of the ATM service, Systinet creates a matching WSDL and stores it at `/services/atm/atm2.0.wsdl`.
- Systinet expects to find other versions of the ATM WSDL at `/services/atm/atmX.Y.wsdl`

When there is no version in a filename Systinet assumes it is the initial version. For example, if you publish `readme.txt` to the ATM folder it becomes version 1.0 and if you later create a version 2.0 of the ATM service, `/services/atm/readme.txt` is cloned as `/services/atm/readme2.0.txt`.

Systinet includes an XML representation of a version in filename strategy:

```
<versioningSchema xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-filename schema">
```

```
  <masks>
    <mask name="any">^((?:/.*)?/([^/\.]*?))(\d+(?:\.\d+)*)?(\.[^/]*)?$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="any"/>
      <group>3</group>
      <property>location</property>
    </version>
    <other>
      <maskref ref="any"/>
      <group>1</group>
      <property>location</property>
      <match>$1$4</match>
    </other>
  </source>
  <location>
    <maskref ref="any"/>
    <version>
      <group>3</group>
    </version>
  </location>
</versioningSchema>
```

"Configuring Versioning" on page 231 describes the elements of this strategy in more detail.

Version in Folders

A common alternative strategy is to group resources with the same version number in the same folder.

In this case, Systinet versions in the following way:

- The WSDL defining version 1.0 of the ATM service exists at `/services/atm/v1.0/atm.wsdl`
- When you create version 2.0 of the ATM service, Systinet creates a matching WSDL and stores it at `/services/atm/v2.0/atm.wsdl`.
- Systinet expects to find other versions of the ATM WSDL at `/services/atm/vX.Y/atm.wsdl`

When there is no version found in a folder containing published content Systinet assumes it is the initial version. For example, if you publish `readme.txt` to the ATM folder it becomes version 1.0 and if you later create a version 2.0 of the ATM service, `/services/atm/readme.txt` is cloned as `/services/atm/v2.0/readme.txt`.

Systinet includes an XML representation of a version in folder strategy:

```
<versioningSchema xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-folder schema">
  <masks>
    <mask name="vif">^(.*?)(/v(\d+(?:\.\d+)*))?(/[^/]+)$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="vif"/>
      <group>3</group>
      <property>location</property>
    </version>
    <other>
      <maskref ref="vif"/>
      <group>1</group>
      <property>location</property>
      <match>$1$4</match>
    </other>
  </source>
  <location>
    <maskref ref="vif"/>
    <version>
      <group prefix="/v">2</group>
    </version>
  </location>
</versioningSchema>
```

Version in Namespaces and Folders

A more advanced strategy is to use the convention that stores the version of a WSDL artifact in its `targetNamespace` and the containing folder.

In this case, Systinet versions in the following way:

- The WSDL defining version 1.0 of the ATM service exists at `/services/atm/v1.0/atm.wsdl` and its namespace is `http://my.company.com/services/atm/v1.0`.
- When you create version 2.0 of the ATM service, Systinet creates a matching WSDL and stores it at `/services/atm/v2.0/atm.wsdl` and changes its namespace to `http://my.company.com/services/atm/v2.0`.
- Systinet expects to find other versions of the ATM WSDL at `/services/atm/vX.Y/atm.wsdl` and to have namespace `http://my.company.com/services/atm/vX.Y`.

Systinet includes an XML representation of a version in namespace and folder strategy:

```
<versioningSchema xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-namespace-and-folder schema">
  <masks>
    <mask name="vif">^(.*?)(/v(\d+(?:\.\d+)*))?(/[^/]+)$</mask>
    <mask name="vns">^(.*/v)(\d+(?:_\d+)*)$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="vns"/>
      <group separator="_">2</group>
      <property>targetNamespace</property>
    </version>
    <other>
      <maskref ref="vif"/>
      <group>1</group>
      <property>location</property>
      <match>$1$4</match>
    </other>
  </source>
  <location>
    <maskref ref="vif"/>
    <version>
      <group prefix="/v">2</group>
    </version>
  </location>
</versioningSchema>
```

Caution: Only apply this strategy for artifacts that have the `targetNamepace` property defined (for example, WSDLs and XSDs). This strategy can obtain the version from just the folder location, but by binding the version to the namespace, Systinet automatically updates the namespace in the new version.

Version in Namespaces and Filenames

Another advanced strategy is to use the convention that stores the version of a WSDL artifact in its `targetNamespace` and the filename.

- When publishing documents (WSDL, XSD), e.g. `Hello.wsdl`, containing target namespace `http://my.company.com/services/atm/v1_0` (v1_0 not v1.0):
  - Systinet will follow the regular expression and automatically create an artifact having version 1.0 (*)
  - This artifact has location /defaultDomain/Hello.wsdl (Systinet keeps the original file name)
- From artifact version 1.0 (*), when creating a new version 2.0 of this artifact:
  - The new one is stored at `/defaultDomain /Hello2.0.wsdl` automatically
  - `Hello2.0.wsdl` will have the target namespace `http://my.company.com/services/atm/v2_0` (v2_0 not v2.0):
- `versionInNamespaceAndFilenameSchema.xml`:
  - this file contains a regular expression
  - Systinet will follow that to parse a version in targetNamespace and specify the fileName for storage in the repository.

- Based on this, Systinet creates a new version of an artifact (WSDL,XSD) (new targetNamespace, new location)

```
<versioningSchema xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-namespace-and-filename schema">
  <masks>
    <mask name="any">^((?:/.*)?/([^/\.]*?))(\d+(?:\.\d+)*)?(\.[^/]*)?$</mask>
    <mask name="vns">^(.*/v)(\d+(?:_\d+)*)$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="vns"/>
      <group separator="_">2</group>
      <property>targetNamespace</property>
    </version>
    <other>
      <maskref ref="any"/>
      <group>1</group>
      <property>location</property>
      <match>$1$4</match>
    </other>
  </source>
  <location>
    <maskref ref="any"/>
    <version>
      <group>3</group>
    </version>
  </location>
</versioningSchema>
```

Custom Versioning Strategies

You can devise your own strategy which should consist of the following information:

- The property to use to extract the version number. For example, location or targetNamespace.
- The regular mask to apply for version extraction.
- The property to use to look up other versions.
- The regular mask to apply to look up other versions.
- The regular mask that specifies the specific position of the version string in the content location.

HP recommends basing any customization on one of the predefined strategies, preferably the one closest to the intended behaviour. For specific requirements that cannot be expressed as regular expressions, it is possible to provide a custom implementation of the versioning schema.

For details of some more advanced strategies, see "Advanced Versioning" on page 241.

Enabling Versioning Strategies

Systinet enables you to apply an individual versioning strategy for specified artifact types and a default strategy for the remaining unspecified artifact types. In this way you can create a complex strategy based on various artifact types and data content.

To enable a customized versioning strategy, you must change the system configuration.

To enable a versioning strategy:

1. Stop the application server.

2. Copy the versioning schema you want to use from `SYSTINET_HOME/deploy/initial/hp-soa-systinet.ear/lib/sc-versioning.jar/xml/`.

3. Modify the schema as required.

4. Export the configuration with the command:

   SYSTINET_HOME/bin/export --image *IMAGE* --components configuration

5. Add your modified schema to a new ***IMAGE***`/configuration/xml/` folder.

6. Open *IMAGE*`/configuration/configuration-properties.xml` in a text editor.

7. To specify a versioning strategy for a specific artifact type add the following property:

   ```
   <property name="platform.versioning.schema.profile.SDM_NAME"
   ref="PATH/FILENAME.xml" type="bin"/>
   ```

   Where SDM_NAME is the local name of the artifact type in the SDM model and the file is an XML representation of a versioning strategy.

   The following example applies a folder and namespace strategy to WSDLs:

   WSDL Version Property

   ```
   <property name="platform.versioning.schema.profile.wsdlArtifact"
       ref="xml/versionInFolderAndNamespaceSchema.xml"
       type="bin"/>
   ```

8. To specify a versioning strategy for the remaining unspecified versionable artifact types add the following property:

   ```
   <property name="platform.versioning.schema.default" ref="xml/FILENAME.xml"
   type="bin"/>
   ```

   Where the file is an XML representation of a versioning strategy.

   The following example applies a folder strategy to unspecified artifact types:

   Default Version Property

```
<property name="platform.versioning.schema.default"
    ref="versionInFolderSchema.xml"
    type="bin"/>
```

9. Save `configuration-properties.xml`.

10. Upload the new configuration with:

    SYSTINET_HOME/bin/import --image *image* --components configuration --configuration-application --configuration-system

11. Restart the application server.

Systinet includes the following predefined versioning strategies.

- `versionInFilenameSchema.xml` (default)

  For details, see "Version in Filenames" on page 232.

- `versionInFolderSchema.xml`

  For details, see "Version in Folders" on page 233.

- `versionInNamespaceAndFolderSchema.xml`

  For details, see "Version in Namespaces and Folders" on page 234.

- `versionInFolderAndFilenameSchema.xml`

  For details, see "Version in Folder and Filename" on page 241.

- `versionInAnyFolderSchema.xml`

  For details, see "Version in Any Folder" on page 243.

- `versionInAnyFolderOrFilenameSchema.xml`

  For details, see "Version in Any Folder or Filename" on page 244.

- `versionInFilenameOrAnyFolderSchema.xml`

  For details, see "Configuring Versioning" on page 231.

- `versionInAnyFilenameAndFolderSchema.xml`

  For details, see "Version in Any Folder and Filename" on page 246.

To enable one of these strategies you do not need to provide an XML file. Add properties to the configuration file in the following formats:

```
<property name="platform.versioning.schema.default" value="xml/FILENAME.xml"/>
```

```
<property name="platform.versioning.schema.profile.SDM_NAME"
value="xml/FILENAME.xml"/>
```

Where the file is one of the provided versioning strategy files.

Default Versioning Schema

Systinet uses version in Namespace and Filename as its default versioning strategy as described in "Version in Namespaces and Filenames" on page 235. This section looks at this strategy in more detail and describes the following elements:

- "Versioning Schema Element" below
- "Masks Element" below
- "Version Element" on the next page
- "Other Element" on the next page
- "Location Element" on page 241

Versioning Schema Element

The top level `versioningSchema` element contains (besides the namespace declaration) the `name` attribute. This has no impact on the versioning strategy and just provides a readable description.

```
<versioningSchema
    xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-filename schema">
  ...
</versioningSchema>
```

Masks Element

The `masks` element provides the definition of named regular expressions that are referenced by other elements.

```
<masks>
  <mask
      name="any">^((?:/.*)?/([^/\.]*?))(\d(?:\.\d)*)?(\.[^/]*)?$</mask>
</masks>
```

In this strategy, there is a single expression called `any`. The syntax uses Java regular expressions (from `java.util.regex`).

The `any` mask matches the following folder and filename elements:

- An arbitrary folder structure (represented by capturing group 1).
- The filename (a concatenation of groups 2, 3, and 4) which possibly contains the following:
    - A number-dot-number segment (group 3).
    - A (multiple) file extension (group 4).

Version Element

After the mask element, the schema contains a source element containing a version sub-element.

```
<source>
  <version>
    <maskref ref="any"/>
    <group>3</group>
    <property>location</property>
  </version>
  ...
</source>
```

The version element specifies how to extract the version by providing the following information:

- The name of the property that the version is based on. In this case, location.
- The mask to apply to the value of that propety. In this case, any.
- The group that represents the dissected location. In this case, group 3, the number-dot-number part of the mask corresponds to the version.

Other Element

The next part of the source element is the other sub-element.

```
<source>
  ...
  <other>
    <maskref ref="any"/>
    <group>1</group>
    <property>location</property>
    <match>$1$4</match>
  </other>
</source>
```

The element has a similar structure to the version element and specifies how to find other versions of the same resource.

- The name of the property to use for identification. In this case, the location property.
- The mask to apply to the value of that property. In this case, the any mask.
- The group that represents that dissected location common for all siblings. In this case, group 1, the location path preceding the number-dot-number seqment.

Unless otherwise specified (by the operator attribute of the other element), the common part is handled as a prefix.

For example, /atm/readme1.txt and /atm/readme2.txt are considered as siblings as they have a common /atm/readme prefix. However, this behaviour is insufficient as it allows the file extension to vary.

The match element specifies a regular expression replacement to apply to the mask:

- Files are only considered as siblings if the properties with the mask and match replacement are equal.
- Both the group and the operator are considered when Systinet performs its search and both must be specified appropriately.

With the mask and replacement Systinet does not consider `/atm/readme1.txt` and `/atm/readme2.doc` to be siblings.

Location Element

The remaining element is `location` which controls the placement of the content in the repository when Systinet creates a new version.

```
<location>
  <maskref ref="any"/>
  <version>
    <group>3</group>
  </version>
</location>
```

The mask is applied to the location of the existing content which is the basis of the new content. In this case, `any`.

`version` contains a list of groups in the mask that contain versioning information. In this case, group 3, the number-dot-number segment.

When creating a version 2, of `/atm/readme1.txt`, Systinet replaces the number-dot-number segment of the existing location with the new version value, giving `/atm/readme2.txt`.

Advanced Versioning

This section describes more advanced versioning strategies with reference to the schema details described in "Configuring Versioning" on page 231.

The following advanced strategies are available and are available in Systinet for reference in the configuration file.

- "Version in Folder and Filename" below
- "Version in Any Folder" on page 243
- "Version in Any Folder or Filename" on page 244
- "Configuring Versioning" on page 231
- "Version in Any Folder and Filename" on page 246

"Versioning Notes" on page 247 provides some information that you should take into account if you decide to implement your own custom versioning strategy.

Version in Folder and Filename

This strategy demonstrates the possibility to project the version into both folder and file name.

The important elements are the following:

- The any mask allows the version to be present in both the folder and filename.

```
<masks>
  <mask name="any">^(/.*?)?(/v(\d+(?:\.\d+)*))?/([^/\.]*?)
      (\d+(?:\.\d+)*)?(\.[^/]*)?$</mask>
</masks>
```

- This corresponds to two groups in the version element.

```
<version>
  <maskref ref="any"/>
  <group>3</group> <!-- corresponds to version in folder -->
  <group>5</group> <!-- corresponds to version in filename -->
  <property>location</property>
</version>
```

- Systinet checks the version in the folder first. If it is not present, Systinet checks the filename. When Systinet creates a new version, the placement of version is retained.

```
<location>
  <maskref ref="any"/>
  <version>
    <group mandatory="false" prefix="/v">2</group>
    <!-- if there is no version in folder name it is not injected
        (mandatory=false), when injecting, /v is prepended to
        version (this complements /v in the source mask)-->
    <group mandatory="false" primary="true">5</group>
    <!-- if there was no version in file name it won't be injected
        (mandatory=false), however if version wasn't found anywhere,
        it will be injected there (primary=true)-->
  </version>
</location>
```

Systinet includes an XML representation of a version in folder and filename strategy:

```
<versioningSchema
    xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-folder-and-filename schema">
  <masks>
    <mask name="any">^(/.*?)?(/v(\d+(?:\.\d+)*))?/([^/\.]*?)
        (\d+(?:\.\d+)*)?(\.[^/]*)?$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="any"/>
      <group>3</group>
      <group>5</group>
      <property>location</property>
    </version>
    <other>
      <maskref ref="any"/>
      <group>1</group>
      <property>location</property>
      <match>$1$4$6</match>
    </other>
  </source>
  <location>
    <maskref ref="any"/>
    <version>
      <group mandatory="false" prefix="/v">2</group>
      <group mandatory="false" primary="true">5</group>
    </version>
  </location>
</versioningSchema>
```

Version in Any Folder

This strategy is a variation on the version in folder strategy described in "Version in Folders" on page 233. However, in this strategy, the version may be present anywhere in the folder hierarchy, with preference given to the folder that most closely matches the filename.

The important elements are the following:

- To achieve this strategy an `OR` is introduced to the regular expression.

  ```
  <masks>
    <!-- (matches if version is given) OR
        (matches if version is not given) -->
    <mask name="vif">^(?:(.*)(/v(\d+(?:\.\d+)*))(/.+))|
        (?:(.*)(/v(?:\d+(?:\.\d+)*))?(/[^/]+))$</mask>
  </masks>
  ```

- To reflect this `OR` when looking for sibling resources, multiple groups appear in the `<other>` element.

  ```
  <other>
    <maskref ref="vif"/>
    <!-- if version is given, preceding path is used for lookup -->
    <group>1</group>
    <!-- if version is not given, path except filename is used-->
    <group>5</group>
    <property>location</property>
    <match>$1$4$5$7</match>
    <!-- both variants are listed, either $1$4 or $5$7 are empty
        (depending on whether the version folder is or is not present) -->
  </other>
  ```

Systinet includes an XML representation of a version in any folder:

```
<versioningSchema
```

```
    xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-any-folder schema">
  <masks>
    <!-- (version given in any folder)|(version not given) -->
    <mask name="vif">^(?:(.*)(/v(\d+(?:\.\d+)*))(/.+))|
        (?:(.*)(/v(?:\d+(?:\.\d+)*))?(/[^/]+))$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="vif"/>
      <group>3</group> <!-- version in folder -->
      <property>location</property>
    </version>
    <other>
      <maskref ref="vif"/>
      <group>1</group> <!-- prefix if version is explicitly given -->
      <group>5</group> <!-- prefix if version not given -->
      <property>location</property>
      <match>$1$4$5$7</match>
      <!-- both variants are listed, either $1$4 or $5$7 is empty -->
    </other>
  </source>
  <location>
    <maskref ref="vif"/>
    <version>
      <!-- only if version explicitly given it will be replaced -->
      <group mandatory="false" prefix="/v">2</group>
      <!-- if version not given, extra folder is placed at the
          end of the path -->
      <group primary="true" mandatory="false" prefix="/v">6</group>
    </version>
  </location>
</versioningSchema>
```

Version in Any Folder or Filename

This strategy uses the version from any folder, with preference given to the folder most closely matching the filename. If no folder specifies the version, then the filename is used.

```
<versioningSchema
    xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-any-folder-or-filename schema">
  <masks>
    <!-- (version given in any folder)|
        (version possibly in filename) -->
    <mask name="vif">^(?:(.*)(/v(\d+(?:\.\d+)*))(/.+))|
        (?:(.*/[^/\.]*?)(\d+(?:\.\d+)*)?(\.[^/]*)?)$</mask>
  </masks>
  <source>
```

```
    <version>
      <maskref ref="vif"/>
      <group>3</group> <!-- version in folder -->
      <group>6</group> <!-- version in filename -->
      <property>location</property>
    </version>
    <other>
      <maskref ref="vif"/>
      <group>1</group> <!-- prefix if version in folder -->
      <group>5</group> <!-- prefix if version in filename-->
      <property>location</property>
      <match>$1$4$5$7</match>
      <!-- both variants are listed, either $1$4 or $5$7 is empty -->
    </other>
  </source>
  <location>
    <maskref ref="vif"/>
    <version>
      <!-- only if version explicitly given it will be
          replaced in folder -->
      <group mandatory="false" prefix="/v">2</group>
      <!-- if not explicitly given anywhere version is
        placed in filename -->
      <group primary="true" mandatory="false">6</group>
    </version>
  </location>
</versioningSchema>
```

Version in Filename or Any Folder

This strategy uses the version in the filename. If not specified by the filename, then any folder is used, with preference given to the folder most closely matching the filename.

```
<versioningSchema
    xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-filename-or-any-folder schema">
  <masks>
    <!-- (version given in filename)|(version given in folder)|(version not in
filename) -->
    <mask name="vif">^(?:(.*/[^/\.]*?)(\d+(?:\.\d+)*)(\.[^/]*)?)|
        (?:(.*)(/v(\d+(?:\.\d+)*))(/.+))|
        (?:(.*/[^/\.]*?)(\d+(?:\.\d+)*)?(\.[^/]*)?)$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="vif"/>
      <group>2</group> <!-- version in filename -->
      <group>6</group> <!-- version in folder -->
      <property>location</property>
    </version>
    <other>
      <maskref ref="vif"/>
      <group>1</group> <!-- if version is in filename-->
      <group>4</group> <!-- if version is in folder -->
      <group>8</group> <!-- if version not given at all -->
      <property>location</property>
      <match>$1$3$4$7$8$10</match>
      <!-- one of $1$3, $4$7 and $8$10 will be non-empty -->
    </other>
  </source>
  <location>
    <maskref ref="vif"/>
    <version>
      <!-- if version in filename -->
      <group mandatory="false">2</group>
      <!-- if version in folder -->
      <group mandatory="false" prefix="/v">5</group>
      <!-- if version not given at all -->
      <group mandatory="false" primary="true" >9</group>
    </version>
  </location>
</versioningSchema>
```

Version in Any Folder and Filename

This strategy projects the version to both the filename and the folder, with preference to the folder most closely matching the filename.

```
<versioningSchema
    xmlns="http://hp.com/2009/11/systinet/platform/versioning/schema"
    name="Built-in version-in-any-folder-and-filename schema">
  <masks>
    <!-- (version in any folder and possibly in filename)|
        (version possibly in filename) -->
    <mask name="vif">^(?:(.*)(/v(\d+(?:\.\d+)*))(/.*)?(/[^/\.]*?)
        (\d+(?:\.\d+)*)?(\.[^/]*)?)|
        (?:(.*/[^/\.]*?)(\d+(?:\.\d+)*)?(\.[^/]*)?)$</mask>
  </masks>
  <source>
    <version>
      <maskref ref="vif"/>
      <group>3</group> <!-- version in folder -->
      <group>9</group> <!-- version in filename -->
      <property>location</property>
    </version>
    <other>
      <maskref ref="vif"/>
      <group>1</group> <!-- prefix if version in folder -->
      <group>8</group> <!-- prefix if version in filename -->
      <property>location</property>
      <match>$1$4$5$7$8$10</match>
      <!-- all groups except those (possibly) containing version -->
    </other>
  </source>
  <location>
    <maskref ref="vif"/>
    <version>
      <!-- version in folder -->
      <group mandatory="false" prefix="/v">2</group>
      <!-- version in filename (when also in folder) -->
      <group mandatory="false">6</group>
      <!-- version in filename (when not in folder) -->
      <group mandatory="false" primary="true">9</group>
    </version>
  </location>
</versioningSchema>
```

Versioning Notes

While it is possible to devise your own versioning strategy, you must ensure its validity. The basic rule is to ensure that the strategy can handle whatever input is provided.

Take into account the following considerations:

- **Non-conforming content**

  In the default strategy, the `any` mask applied to the location always matches, even if there is no version in the filename because the *number-dot-number* segment is optional and if it is not present, the initial version (1.0) is assumed.

  If the mask defining the placement is restrictive and a resource with a non-matching location is cloned, the versioning schema cannot determine the new location and the cloned resource is placed in a fallback location.

  For example, consider the following strategy for WSDL content:

  ```
  ...
    <masks>
      <mask name="strict">^/wsdls/[^/]+(\d(\.\d)*).wsdl$</mask>
    </masks>
    ...
    <location>
      <maskref ref="strict"/>
      <version>
        <group>1</group>
      </version>
  ```

```
  </location>
...
```

This strategy assumes that WSDL content meets the following criteria:

- The file is always placed in the `/wsdls` folder.
- There is always a number-dot-number segment in the filename.
- The file always has a `.wsdl` extension.

If any of these rules are not met for a WSDL in the repository when creating a version 2.0, Systinet applies the following fallback mechanism:

- `/wsdls/service.wsdl` clones as `/wsdls/service.wsdl_clone_2_0`
- `/wsdls/demo/service1.wsdl` clones as `/wsdls/demo/service1/wsdl_clone_2.0`

- **Property availability**

  The property referenced in both `version` and `other` elements must be defined for the particular artifact type the strategy applies to. If the strategy does not apply to a particular artifact type, then the property must be a common property of all artifacts, for example, location, name, or description.

Chapter 14: Setup Tool

Included with Systinet is the Setup Tool. You can use it for the following functions, which you select as **Scenarios** when running the tool.

To access the Setup Tool user interface, execute the following command:

SYSTINET_HOME/bin/setup

The Setup GUI opens at the Welcome screen.

Click **Next** to be presented a set of scenarios, as described in the following sections:

- "Changing the License Key" below
- "Applying Extensions" on page 215
- "Updating Systinet" on the next page
- "Advanced Setup Tool Options" on the next page

The Setup Tool can also be used in command line mode.

For details, see "Setup Tool Command-Line Options" on the next page.

Note: By default, the Setup Tool does not allow you to import data or apply extensions while there is a server running in order to protect data consistency. In some environments (for example, behind a load balancer proxy or using Siteminder) there is always something running at the server endpoint. To enable the Setup Tool in these environments set the `install.ignore.running.platform` property to `TRUE`. For details, see "How to Manage System Settings" on page 81. Alternatively, execute the Setup Tool with the following command option: **-Dinstall.ignore.running.platform=true**

Changing the License Key

The Setup Tool enables you to change the license.

To Change the License Key with the Setup Tool:

1. In the Scenario Selection page, select **Change License Key**, and then click **Next**.

   The License Information page opens.

2. In the License Information page, do one of the following:

  - Select **Install a 60 day evaluation license**.
  - Select **Enter license details**, and input the license details provided by your sales representative.

  Click **Next**.

3. Click **Next** through each confirmation and progress page, and when the setup is complete, do one of the following:
   - Click **Setup Again** to return to the Scenario Selection page.
   - Click **Finish** to exit the Setup Tool.

> **Note:** The Administration Tab also provides support for changing the license key. For details, see "License Management" on page 78.

Updating Systinet

The Setup Tool enables you to install updates to Systinet, which are downloaded or copied to the `SYSTINET_HOME/updates` directory.

For details, see "Applying Updates" on page 219.

Advanced Setup Tool Options

The Advanced scenario enables you to select specific parts of the configuration procedure to suit the needs of a specific task.

To Select Specific Configuration Processes:

- In the Scenario Selection page, select **Advanced**, and then click **Next**.

  The Custom Scenario Selection page opens and enables you to select which parts of the configuration you want to execute.

Every part of the configuration process is listed as an individual step. The steps required for a particular process vary depending on what configuration you want to change.

Setup Tool Command-Line Options

The Setup Tool can also be executed as a command line tool.

The setup command is:

SYSTINET_HOME/bin/setup [OPTIONS]

The following options are available:

- **-h, --help [scenarios|steps]**

  Display the available options or list the available scenarios or steps in the console.

- **-c, --console**

  Execute the Setup Tool in console mode.

- **-a, --dbadmin-mode**

  Enables DB administrator mode. The setup stops after creating the DB scripts, allowing the administrator to execute them manually. Continue installation after script execution with **setup -c**.

- **-n, --scenarios *SCENARIO***

  Execute only the specified steps in the installation. Use **--help scenarios** to view a list of available scenarios.

- **-p, --steps [*comma separated list of steps*]**

  Execute only the specified steps in the installation. Use **--help steps** to view a list of available steps.

- **-u, --use-config *FILE***

  Use the properties in the specified file to override the default or current configuration properties.

- **--passphrase *PASSPHRASE***

  If password encryption is enabled, specify the passphrase to use for encryption.

- **-d, --debug**

  Execute the setup in debug mode. All properties, SQL statements, and installation details are output to `SYSTINET_HOME/log/setup.log`.

- **--silent**

  For some automatic scripts, adding warning to Apply Extension step could break the script. The solution is adding silent mode to setup command.

Chapter 15: Administration Utilities

Systinet administration utilities consist of command-line tools located in the `bin` directory of the installation folder.

The utilities are summarized in "Systinet Utilities" below.

This chapter describes the following utilities:

- "Export Tool" on the next page
- "Import Tool" on page 256
- "Migration Tool" on page 265
- "Rebrand Tool" on page 265
- "Reset Tool" on page 266
- "SDM to Database Mapping Tool" on page 267
- "SSL Tool" on page 267

Note: If passwords are encrypted, set the option `--passphrase passphrase` on the command-line when you launch any tool that requires authentication.

Systinet Utilities

The Systinet utilities are located in `SYSTINET_HOME/bin`. These are either batch `BAT` files or shell `SH` scripts, depending on the server operating system.

Note: If a utility is not in `SYSTINET_HOME/bin`, a relative path is shown for the command in table "Admin Utilities" below.

Admin Utilities

Command	Description
env	A script used by other Systinet tools to set system variables. Do not execute this script directly.
env-jboss	Called by serverstart to set system variables for the application server. Do not execute this script directly.

Command	Description
export	Creates a data image for specified components of Systinet. For details, see "Export Tool" below.
import	Imports a data image for specified components of Systinet. For details, see "Import Tool" on page 256.
migrate	Converts an image of the repository data from Systinet 4.x to an image compatible with this version. For details, see "Migration Tool" on page 265. Use **--help** to view the available options.
rebrand	Re-brand or co-brand Systinet with your company specific logos and names. For details, see "Rebrand Tool" on page 265.
reset	Resets the data for specified components of Systinet. For details, see "Reset Tool" on page 266.
../lib/sdm/bin/sdm2dbmap	Creates a report of the relationship between the SDM structure and the database tables. For details, see "SDM to Database Mapping Tool" on page 267.
serverstart	Calls env-jboss to set critical system variables for JBoss, and then starts the platform application server. For other application servers, use the server start functionality in the application server.
serverstop	Stops the platform application server for JBoss. For other application servers, use the server stop functionality in the application server.
setup	Starts the Setup Tool to reconfigure the platform server. For details, see "Setup Tool" on page 249. Use **--help** to view the available options.
ssltool	Configures and views your SSL configuration. For details, see "SSL Tool" on page 267.

Note: If a command requires arguments, running it without arguments displays a help screen, unless otherwise stated.

Export Tool

The **export** command enables you to export the configuration and data in the database to an image, and then import that data at a later date.

The syntax for export is:

export --image *IMAGE_NAME* [OPTIONS]

The Export Tool includes the following options:

`--image` *`IMAGE_NAME`*	The path to the directory where the image is stored.
`--components [COMPONENT]`	The following component options are available: - `all` This is also the default if you omit `--components`. Exports all the configurations and data except the license. - `configuration` The configuration data. - `content` All data without the configuration and security data. - `security` The security configuration. User profiles, groups, and roles and the default ACLs for newly created resources.
`--quiet`	Execute the command without a confirmation request.
`--silent`	Less verbose console output.
`--last-revision`	Exports only last artifact revision. Last revision number will be reset to 1.
`--include-generated-data`	Enforce export of all non-valuable data such as user event, execution, and validation data. This option also covers these options: - `--userEvents-dontSkip` - `--executions-flat-limit` - `--policyManager-validations`
`--executions-flat-limit` *`N`*	Export execution reports in a flat layout with a limit of *`N`* reports/items. **Note:** By default, execution reports are exported in a flat layout up to a predefined number of reports and items. If the limit is exceeded a hashed layout is used instead to prevent potential excess directory errors. Use this command to define your own limit for flat layout exports.

`--configuration-all`	Export the complete configuration (not recommended).
`--configuration-application`	Export the application configuration (default).
`--configuration-system`	Export the system configuration (default).
`--configuration-licence`	Export license details.
`--policyManager-validations`	Enable the export of policy manager validation data (not exported by default).
`--userEvents-dontSkip`	Export user event data (not exported by default).

Caution: Systinet must not be running when you execute these commands.

The export creates the directory specified by `IMAGE_NAME`, containing the following, depending on the component options used:

- `image.properties`

  A file containing the export execution properties and a list of the data sets exported.

- `configuration`

  A directory containing the configuration properties files, including role-based UI customizations. The file also contains the license details if you use the **--configuration-license** option.

- `dist`

  Contains properties specific to a particular distribution. Create this data with the **--configuration-force** option.

- `executions`

  A directory containing the execution report results of asynchronous tasks such as publishing, discovery, and bulk operations.

- `lifecycle`

  A directory containing the lifecycle data.

- `platform`

  A directory containing the service catalog data.

- `policyManager`

  A directory containing the policy data.

- `reporting`

A directory containing the reporting definitions.

- `security`

  A directory containing the security configuration.

- `userEvents`

  A directory user specific event information.

Import Tool

The **import** command enables you to import Systinet configuration and data to the database from an image.

> **Note:** When importing an image with governed artifacts using the default options, lifecycle processes with identical Uuids in the image are not imported. To avoid inconsistencies in the lifecycle of artifacts use the import tool with the **--reset** option which deletes all existing artifacts and lifecycle processes.

> **Note:** Importing a Systinet 10.01 image replaces the current group membership with group membership from the image. Importing an image migrated from Systinet 4.x, merges the imported group membership with the current group membership.

The syntax for import is:

import --image *IMAGE_NAME* [OPTIONS]

> **Note:** HP recommends updating Oracle Database schema statistics after importing large amounts of data. Old statistics may impact the performance of some data queries. Consult your database administrator.

> **Note:** Systinet provides some OOTB HTTP Status Code artifacts and HTTP Header artifacts which can be used while publishing WADL or Swagger files. The new HTTP Status Code or HTTP Header artifact will be created only when its code number or header value is found in the existing list.

HTTP Status Code Artifacts

The following is the list of OOTB HTTP Status code artifacts:

HTTP Status Codes	Description

100 Continue	This means that the server has received the request headers, and that the client should proceed to send the request body
101 Switching Protocols	This means the requester has asked the server to switch protocols and the server is acknowledging that it will do so.
102 Processing	This code indicates that the server has received and is processing the request, but no response is available yet.This prevents the client from timing out and assuming the request was lost.
200 OK	Standard response for successful HTTP requests.
201 Created	The request has been fulfilled and resulted in a new resource being created.
202 Accepted	The request has been accepted for processing, but the processing has not been completed.
203 Non-Authoritative Information (since HTTP/1.1)	The server successfully processed the request, but is returning information that may be from another source.
204 No Content	The server successfully processed the request, but is not returning any content. Usually used as a response to a successful delete request.
205 Reset Content	The server successfully processed the request, but is not returning any content. Unlike a 204 response, this response requires that the requester reset the document view.
206 Partial Content	The server is delivering only part of the resource (byte serving) due to a range header sent by the client. The range header is used by tools like wget to enable resuming of interrupted downloads, or split a download into multiple simultaneous streams.
207 Multi-Status	The message body that follows is an XML message and can contain a number of separate response codes, depending on how many sub-requests were made.
208 Already Reported	The members of a DAV binding have already been enumerated in a previous reply to this request, and are not being included again.
226 IM Used	The server has fulfilled a request for the resource, and the response is a representation of the result of one or more instance-manipulations applied to the current instance.
300 Multiple Choices	Indicates multiple options for the resource that the client may follow.
301 Moved Permanently	This and all future requests should be directed to the given URI.
302 Found	This is an example of industry practice contradicting the standard.

303 See Other	The response to the request can be found under another URI using a GET method.
304 Not Modified	Indicates that the resource has not been modified since the version specified by the request headers If-Modified-Since or If-None-Match.
305 Use Proxy	The requested resource is only available through a proxy, whose address is provided in the response.
306 Switch Proxy	No longer used. Originally meant "Subsequent requests should use the specified proxy.
307 Temporary Redirect	In this case, the request should be repeated with another URI; however, future requests should still use the original URI.
308 Permanent Redirect	The request, and all future requests should be repeated using another URI.
400 Bad Request	The server cannot or will not process the request due to something that is perceived to be a client error.
401 Unauthorized	Similar to "403 Forbidden" below, but specifically for use when authentication is required and has failed or has not yet been provided.
402 Payment Required	Reserved for future use. The original intention was that this code might be used as part of some form of digital cash or micropayment scheme, but that has not happened, and this code is not usually used.
403 Forbidden	The request was a valid request, but the server is refusing to respond to it. Unlike a "401 Unauthorized" above response, authenticating will make no difference.
404 Not Found	The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible.
405 Method Not Allowed	A request was made of a resource using a request method not supported by that resource; for example, using GET on a form which requires data to be presented via POST, or using PUT on a read-only resource.
406 Not Acceptable	The requested resource is only capable of generating content not acceptable according to the Accept headers sent in the request.
407 Proxy Authentication Required	The client must first authenticate itself with the proxy.
408 Request Timeout	The server timed out waiting for the request.
409 Conflict	Indicates that the request could not be processed because of conflict in the request, such as an edit conflict in the case of multiple updates.
410 Gone	Indicates that the resource requested is no longer available and will not be available again.

411 Length Required	The request did not specify the length of its content, which is required by the requested resource.
412 Precondition Failed	The server does not meet one of the preconditions that the requester put on the request.
413 Request Entity Too Large	The request is larger than the server is willing or able to process.
414 Request-URI Too Long	The URI provided was too long for the server to process.
415 Unsupported Media Type	The request entity has a media type which the server or resource does not support.
416 Requested Range Not Satisfiable	The client has asked for a portion of the file (byte serving), but the server cannot supply that portion.
417 Expectation Failed	The server cannot meet the requirements of the Expect request-header field.
418 I'm a teapot	This code was defined in 1998 as one of the traditional and is not expected to be implemented by actual HTTP servers.
419 Authentication Timeout	Not a part of the HTTP standard, 419 Authentication Timeout denotes that previously valid authentication has expired.
420 Method Failure	Not part of the HTTP standard, but defined by Spring in the HttpStatus class to be used when a method failed. This status code is deprecated by Spring.
420 Enhance Your Calm	Not part of the HTTP standard, but returned by version 1 of the Twitter Search and Trends API when the client is being rate limited.
422 Unprocessable Entity	The request was well-formed but was unable to be followed due to semantic errors.
423 Locked	The resource that is being accessed is locked.[
424 Failed Dependency	The request failed due to failure of a previous request.
426 Upgrade Required	The client should switch to a different protocol such as TLS/1.0, given in the Upgrade header field.
428 Precondition Required	The origin server requires the request to be conditional.
429 Too Many Requests	The user has sent too many requests in a given amount of time.

431 Request Header Fields Too Large	The server is unwilling to process the request because either an individual header field, or all the header fields collectively, are too large.
440 Login Timeout (Microsoft)	A Microsoft extension. Indicates that your session has expired.
444 No Response (Nginx)	Used in Nginx logs to indicate that the server has returned no information to the client and closed the connection
449 Retry With (Microsoft)	A Microsoft extension. The request should be retried after performing the appropriate action.
450 Blocked by Windows Parental Controls (Microsoft)	A Microsoft extension. This error is given when Windows Parental Controls are turned on and are blocking access to the given webpage.
451 Unavailable For Legal Reasons (Internet draft)	Defined in the internet draft "A New HTTP Status Code for Legally-restricted Resources".
451 Redirect (Microsoft)	Used in Exchange ActiveSync if there either is a more efficient server to use or the server cannot access the users' mailbox.
494 Request Header Too Large (Nginx)	Nginx internal code similar to 431 but it was introduced earlier in version 0.9.4 (on January 21, 2011).
495 Cert Error (Nginx)	Nginx internal code used when SSL client certificate error occurred to distinguish it from 4XX in a log and an error page redirection.
496 No Cert (Nginx)	Nginx internal code used when client didn't provide certificate to distinguish it from 4XX in a log and an error page redirection.
497 HTTP to HTTPS (Nginx)	Nginx internal code used for the plain HTTP requests that are sent to HTTPS port to distinguish it from 4XX in a log and an error page redirection.
498 Token expired/invalid (Esri)	Returned by ArcGIS for Server. A code of 498 indicates an expired or otherwise invalid token.
499 Client Closed Request (Nginx)	Used in Nginx logs to indicate when the connection has been closed by client while the server is still processing its request, making server unable to send a status code back.
499 Token required (Esri)	Returned by ArcGIS for Server. A code of 499 indicates that a token is required (if no token was submitted).

500 Internal Server Error	A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.
501 Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfil the request.
502 Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server.
503 Service Unavailable	The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary state.
504 Gateway Timeout	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server.
505 HTTP Version Not Supported	The server does not support the HTTP protocol version used in the request.
506 Variant Also Negotiates	Transparent content negotiation for the request results in a circular reference.
507 Insufficient Storage	The server is unable to store the representation needed to complete the request.
508 Loop Detected	The server detected an infinite loop while processing the request.
509 Bandwidth Limit Exceeded	This status code is not specified in any RFCs. Its use is unknown.
510 Not Extended	Further extensions to the request are required for the server to fulfill it.
511 Network Authentication Required	The client needs to authenticate to gain network access. Intended for use by intercepting proxies used to control access to the network.
598 Network read timeout error	This status code is not specified in any RFCs, but is used by Microsoft HTTP proxies to signal a network read timeout behind the proxy to a client in front of the proxy.
599 Network connect timeout error	This status code is not specified in any RFCs, but is used by Microsoft HTTP proxies to signal a network connect timeout behind the proxy to a client in front of the proxy.

HTTP Header Artifacts

Name	Header Value	Owner	Domain
Content-Type	application/json,	HP Software Developer	HP Software Products

Content-Type	application/xml	Administrator	Top-level Domain
Content-Type	application/x-www-form-urlencoded	Administrator	Top-level Domain
Content-Type	multipart/form-data	Administrator	Top-level Domain

To Update Oracle Schema Statistics:

- Execute the following command:

  EXEC DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '&1',no_invalidate => FALSE,options => 'GATHER');

  This command does not require database admin privileges and can be run by the schema owner (ownname).

The Import Tool includes the following options:

`--image` *`IMAGE_NAME`*	The path to the directory where the image is stored.
`--validate`	Validate the image before it is imported.
`--validate-only`	Perform only image validation, do not perform import.
`--components [COMPONENT]`	The following component options are available: • `all` This is also the default if you omit `--components`. Imports all the data without the configuration. • `configuration` The configuration data. • `content` All data without the configuration and security data. • `security` The security configuration. User profiles and groups and the default ACLs for newly created resources.
`--quiet`	Execute the command without a confirmation request.
`--silent`	Less verbose console output.

`--reset`	Executes the **reset** command first deleting all existing artifacts and lifecycle processes, with matching component options.
`--force`	If an imported service catalog resource is already in the database, it is overwritten.
`--passphrase` *PASSPHRASE*	Use when the product is installed with Password Encryption enabled. Specify the master passphrase entered during installation or setup.
Platform specific options:	
`--platform-bootstrap`	Import the service catalog data in bootstrap format.
`--platform-update-blacklist`	Append imported service catalog resources to the migration blacklist. Useful for bootstrap installation.
`--platform-rest-blacklist`	Save imported service catalog resources to the migration blacklist.
`--platform-ignore-sdm-merge-warn`	Continue service catalog data import if the SDM merge check only reports warnings.
Reporting specific options:	
`--reporting-reset-blacklist`	Save imported resources to the migration blacklist.
`--executions-force`	If an imported execution report or item is already in the database, it is overwritten.
Configuration specific options:	
`--configuration-all`	Import the complete configuration (not recommended).
`--configuration-application`	Import the application configuration (default).
`--configuration-system`	Import the system configuration.
`--configuration-licence`	Import license details.
`--configuration-passphrase PASSPHRASE`	Use the configuration passphrase of image when importing if image was encrypted.
User events specific options:	
`--userEvents-dontSkip`	Import user event data (not imported by default). Existing user event data is deleted before this data imports.

Caution: Systinet must not be running when you execute these commands.

The import checks the directory specified by `IMAGE_NAME`, which contains the following depending on the image:

- `image.properties`

  A file containing the export execution properties and a list of the data sets exported.

- `configuration`

  A directory containing the configuration properties files, including role-based UI customizations. The file also contains the license details if the export used the **--configuration-license** option.

- `dist`

  Contains properties specific to a particular distribution. Use the **--configuration-force** option to import this data.

- `executions`

  A directory containing the execution report results of asynchronous tasks such as publishing, discovery, and bulk operations.

- `lifecycle`

  A directory containing the lifecycle data.

- `platform`

  A directory containing the service catalog data.

- `policyManager`

  A directory containing the policy data.

- `reporting`

  A directory containing the reporting definitions.

- `security`

  A directory containing the security configuration.

- `userEvents`

  A directory user specific event information.

Note: If specific components are specified, the other component folders are ignored. If a specified component is not present, the import fails.

Migration Tool

The migration tool enables you to convert a data image exported from Systinet 4.x to an image compatible for import to Systinet.

The command for migration is:

migrate [-c] [-t task1,...,taskN] [-b] [-m model-changes] -o *OUTPUT* [-k] -i *IMAGE* [-v]

The Migration Tool includes the following options:

-o, --output *OUTPUT*	Specify the name of the resulting migrated data image. You can specify a directory or a ZIP archive.
-m, --model-changes *model-changes*	Specify explicit model changes, overriding the implicit ones. HP recommend consulting HP Support if you need to use this option.
-k, --keep-temp-dir	When outputting to a ZIP archive, use this option to retain the temporary working directory in the file system.
-b, --non-interactive	Run in non-interactive (batch) mode.
-t, --task *task1,...,taskN*	Execute only specific migration tasks. Use **-t help** for more details.
-c, --concurrent	Concurrent processing of the task queue (experimental - may yield improved performance).
--i, --image *IMAGE*	The path to the image archive file or a path to the directory containing the exported data from Systinet 4.x.
-v, --validate	Perform validation on generated / migrated data.

For details on how to use this tool to migrate data from Systinet 4.x, see "Data Migration" on page 212.

Rebrand Tool

The rebrand tool enables you to extract the images, text, and libraries from Systinet used to identify and brand it as an HP product. You can then modify these files and reapply your own corporate image to Systinet.

The command for rebranding is:

rebrand [OPTIONS]

The Rebrand Tool includes the following options:

--extract	Extract the current branding image.
--apply	Apply a new branding image.

--image *IMAGE*	The path to the image archive file (**--apply**) or a path to directory to export the extracted image to (**--extract**).
--imageconf *FILE*	Custom path to create an image configuration XML file (**--extract** only).
--quiet	Execute with no confirmation queries.
--silent	Less verbose console output.
--force	Ignore branding image consistency check errors, for example missing files, (**--extract** option only).
--passphrase *PASSPHRASE*	Specify the master passphrase entered during installation or setup if password encryption is enabled.

For details on how to use this tool to rebrand the product, see "Rebranding Systinet" on page 228.

Caution: Systinet must not be running when you execute these commands.

Reset Tool

The **reset** command enables you to reset the Systinet data in the database and import the default image.

Note: All artifacts and lifecycle processes are deleted prior to the import. This resolves any conflicts with identical artifacts or process UUIDs between existing and imported data.

The syntax for reset is:

reset [OPTIONS]

The Reset Tool includes the following options:

`--components [COMPONENT]`	The following component options are available: • `all` This is also the default if you omit `--components`. Resets all data. • `content` Resets all data excepting security. • `security` Resets only security configuration. User security profile, groups, roles and default ACL's for newly created resources.
`--quiet`	Execute the command without a confirmation request.

Caution: Systinet must not be running when you execute these commands.

SDM to Database Mapping Tool

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

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

To generate the report, execute the following command:

SYSTINET_HOME/lib/sdm/bin/sdm2dbmap

The mapping report is output to the following file:

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

The output consists of the following parts:

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

SSL Tool

The SSL Tool is a combined tool enabling you to setup client-side SSL for a deployed Systinet application. It also enables you to print SSL server certificates, as well as to download the SSL server certificate chain.

The SSL Tool has the following basic actions:

- **serverInfo**

  Prints the SSL requirements for the specified HTTPS URL, and saves the server certificate to a file.

  For details, see "Identifying Server SSL Requirements" on page 223.

- **keystoreEI**

  Exports or imports SSL certificates to the HPSystinett database keystore or truststore.

For details, see "SSL Server Certificate Trust" on page 224 and "Importing Client Certificates for Two-Way SSL" on page 224.

- **customize**

  Change the effective SSL customization.

  For details, see "SSL Customization" on page 225.

The syntax for ssltool is:

SYSTINET_HOME/bin/ssltool [ACTION] [options]

Execute **ssltool** with no action or options to view the help with some examples.

Execute **ssltool [ACTION] --help** to view specific help for each type of action with the available options.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Administration Guide (Systinet 10.01)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to docteam_systinet@hp.com.

We appreciate your feedback!