HP OpenView Select Identity

Workflow Studio Guide

Software Version: 3.0.1

UNIX® (Sun Solaris) and Windows®
Operating Systems

October 2004

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Portions Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved.

Select Identity uses software from the Apache Jakarta Project including:

- Commons-beanutils.
- Commons-collections.
- Commons-logging.
- Commons-digester.
- Commons-httpclient.

- Element Construction Set (ecs).
- Jakarta-poi.
- Jakarta-regexp.
- Logging Services (log4j).

Additional third party software used by Select Identity includes:

- JasperReports developed by SourceForge.
- iText (for JasperReports) developed by SourceForge.
- BeanShell.
- Xalan from the Apache XML Project.
- Xerces from the Apache XML Project.
- Java API for XML Processing from the Apache XML Project.
- SOAP developed by the Apache Software Foundation.
- JavaMail from SUN Reference Implementation.
- Java Secure Socket Extension (JSSE) from SUN Reference Implementation.
- Java Cryptography Extension (JCE) from SUN Reference Implementation.
- JavaBeans Activation Framework (JAF) from SUN Reference Implementation.
- OpenSPML Toolkit from OpenSPML.org.
- JGraph developed by JGraph.
- Hibernate from Hibernate.org.

Trademark Notices

HP OpenView Select Identity is a trademark of Hewlett-Packard Development Company, L.P.

Microsoft, Windows, the Windows logo, and SQL Server are trademarks or registered trademarks of Microsoft Corporation.

Sun™ workstation, Solaris Operating Environment™ software, SPARCstation™ 20 system, Java technology, and Sun RPC are registered trademarks or trademarks of Sun Microsystems, Inc. JavaScript is a trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

This product includes the Sun Java Runtime. This product includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/.

IBM, DB2 Universal Database, DB2, WebSphere, and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

This product includes software provided by the World Wide Web Consortium. This software includes xml-apis. Copyright © 1994-2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/

Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

AMD and the AMD logo are trademarks of Advanced Micro Devices, Inc.

BEA and WebLogic are registered trademarks of BEA Systems, Inc.

VeriSign is a registered trademark of VeriSign, Inc. Copyright © 2001 VeriSign, Inc. All rights reserved.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Support

Please visit the HP OpenView web site at:

http://openview.hp.com/

There you will find contact information and details about the products, services, and support that HP OpenView offers.

You can go directly to the support web site at:

http://support.openview.hp.com/

The support web site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

contents

1

Introduction to Workflow Studio

As described in the *HP OpenView Select Identity Administrator Guide*, a **workflow process** represents the process by which service-access requests are approved and provisioned by HP OpenView Select Identity. Provisioning includes adding, modifying, and removing user accounts.

The complexity of the workflow process can vary widely depending on your provisioning needs. You can simply provision a user by creating the user in Select Identity then pushing the user account to the external resource. Or, provisioning can require multiple Select Identity administrators' approval. The approval process can also rely on external calls to third-party systems or databases.

For example, when an employee is promoted to manager, he needs access to the company's HCM system to manage other employees. To support these newly-acquired responsibilities, the employee must be granted new entitlements and access privileges. Before giving him access to these systems, upper-level management needs to approve the access requests and the employee must be created in the supporting systems. Thus, the workflow process involves retrieving the names of managers, requesting their approval to add the employee to the HCM systems, provisioning the employee's account, and notifying him that he is now authorized to manage others.

Workflow Studio enables you to create a workflow template that represents a provisioning process. A **workflow template** models this process in order to automate the actions that approvers and systems management software must perform. This guide describes how to use Workflow Studio to create workflow templates and the building blocks you will use.

Workflow Templates in Select Identity

Using the Select Identity client, you can assign workflow templates to request events in a Business Relationship. (A Business Relationship is created as part of a Service.) For example, you can assign a simple provisioning template to an add request for self-registration. This template might perform user provisioning and request a single approval. Thus, when a new user requests access to the service, the template is invoked and an administrator must approve the request before the user is added to the supporting systems.

As Select Identity invokes a template, it creates a workflow instance and performs activities as defined in the template. ("Workflow" refers to a workflow instance.) If you create a more complicated workflow, activities might include the following:

- Selecting a list of approvers by specifying a role created on the Admin Roles home page
- Sending email using one of the email templates created on the Notifications home page
- Calling an external system registered with Select Identity on the External Calls home page

You can generate reports to track the status of request events and the workflows that support them. To view reports, specify parameters on the Request Status home page of the Select Identity client. See Reporting on page 71 for more information.

Concepts and Terms

While building workflow templates in Workflow Studio, you will encounter the following terms:

Activities

An activity represents a step in a process that may be traversed when a workflow template is executed. Activities are the core components of workflow templates; the actions defined in activities do the work necessary to provision users. An activity can contain actions, which set variables to be used throughout the workflow, track approvals, start a subworkflow, send email, call external applications, and so on.

Actions

An activity can contain many actions that invoke functions provided by the workflow engine or external applications. For example, you can log information to a file, set a variable to be used later in the workflow, call an external process to provision a user in Select Identity, or store data in a database.

Blocks

To group a set of related activities, you can create a block. Blocks have two purposes: to define information to be shared by a subset of activities (block-level properties) and to provide block-level reporting. For example, you might define a block that submits an approval request, waits for the response, and returns the status of the request to the workflow. In other words, think of a block as a process within a template.

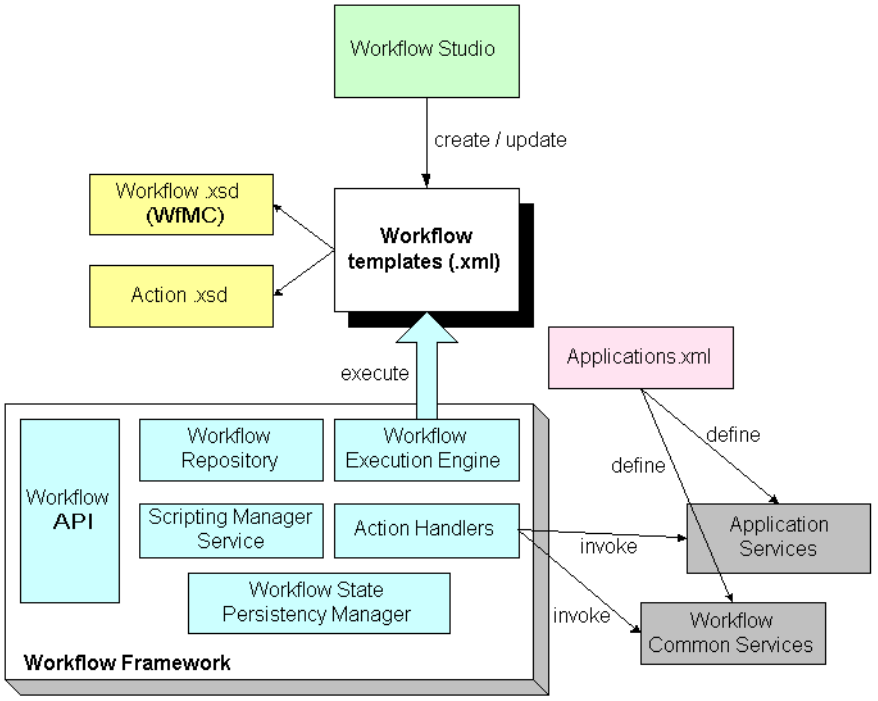
Transitions

A transition provides a link from one activity to another. You can define that one activity always follows another, or you can define a condition that must be met before the workflow transitions from an activity to one or more others. For example, you can define a transition that only allows the workflow to progress if at least two administrators approve a request. If the request is not approved, the workflow can transition to an activity that sends email notification to an administrator.

Behind the Scenes

When you save a workflow template, it is saved in the Select Identity repository as an XML file. Its format is the XML Processing Description Language (XPDL) as defined by the Workflow Management Coalition (WfMC). Every time you click **Save**, a version of the template is saved (a template's XML is never overwritten). This assures that currently running workflow instances do not break because the workflow template changes. (However, you cannot access versions of a template.)

The following diagram illustrates the workflow architecture:



Here is a description of the architecture:

- The Workflow Studio creates templates and stores them as XML files. The `Workflow.xsd` and `Action.xsd` files define the structure and the action data model as defined by WfMC.
- The workflow engine interprets the template using the Workflow Meta Data Model, which defines Select Identity's template structure.

- The Scripting Manager evaluates workflow properties and expressions, and the Workflow State Persistency Manager persists workflow properties and state data.
- The Action Handlers invoke predefined actions (external processes). The `Applications.xml` file defines applications that are referenced by actions in the template. The Workflow Common Services provide common services for use by the engine, and the Application Services interface into Select Identity.
- The Workflow API provides an interface to the query instance state and other information.

Product Documentation

The Select Identity product documentation includes the following:

- Release notes are provided in the top-level directory of the HP OpenView Select Identity CD. This document provides important information about new features included in this release, known defects and limitations, and special usage information that you should be familiar with before using the product.
- For installation and configuration information, refer to the *HP OpenView Select Identity Installation Guide*. All installation prerequisites, system requirements, and procedures are explained in detail in this guide. Specific product configuration and logging settings are included. This guide also includes uninstall and troubleshooting information.
- An *HP OpenView Connector Installation Guide* is provided for each resource connector. These are located on the Select Identity Connector CD.
- Detailed procedures for deployment and system management are documented in the *HP OpenView Select Identity Administrator Guide* and Select Identity online help system. This guide provides detailed concepts and procedures for deploying and configuring the Select Identity system. In the online help system, tasks are grouped by the administrative functions that govern them.

- The *HP OpenView Select Identity Workflow Studio Guide* provides detailed information about using Workflow Studio to create workflow templates. It also describes how to create reports that enable managers and approvers to check the status of account activities.
- The *HP OpenView Select Identity External Call Developer Guide* provides detailed information about creating calls to third-party applications. These calls can then be deployed in Select Identity to constrain attribute values or facilitate workflow processes. In addition, JavaDoc is provided for this API. To view this help, extract the `javadoc.jar` file in the `docs/api_help/external_calls/Javadoc` directory on the HP OpenView Select Identity CD.
- If you need to develop connectors, which enable you to connect to external systems for provisioning, refer to the *HP OpenView Select Identity Connector Developer Guide*. This document provides an overview of the Connector API and the steps required to build a connector. The audience of this guide is developers familiar with Java.

  JavaDoc is also provided for the Connector API. To view this help, extract the `javadoc.jar` file in the `docs/api_help/connectors/Javadoc` directory on the HP OpenView Select Identity CD.
- The *HP OpenView Select Identity Web Service Developer Guide* describes the Web Service, which enables you to programmatically provision users in Select Identity. This guide provides an overview of the operations you can perform through use of the Web Service, including SPML examples for each operation.

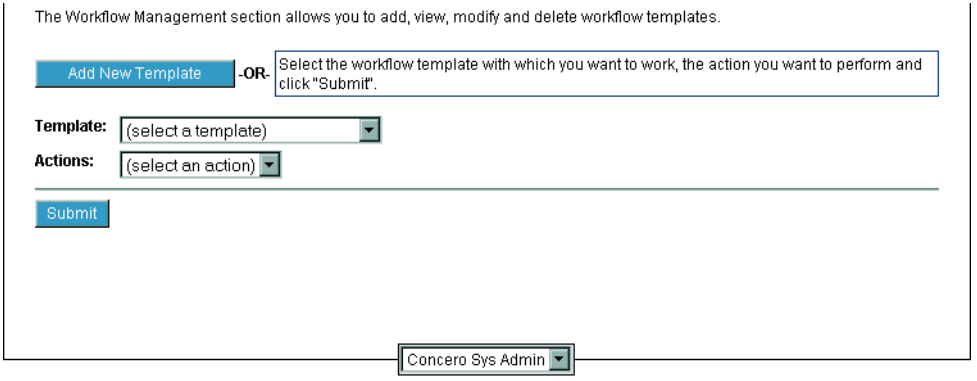
  An independent, web-based help system is available for this API. To view this help, double-click the `index.htm` file in the `docs/api_help/web_service/help` directory on the HP OpenView Select Identity CD.

2

Getting Started

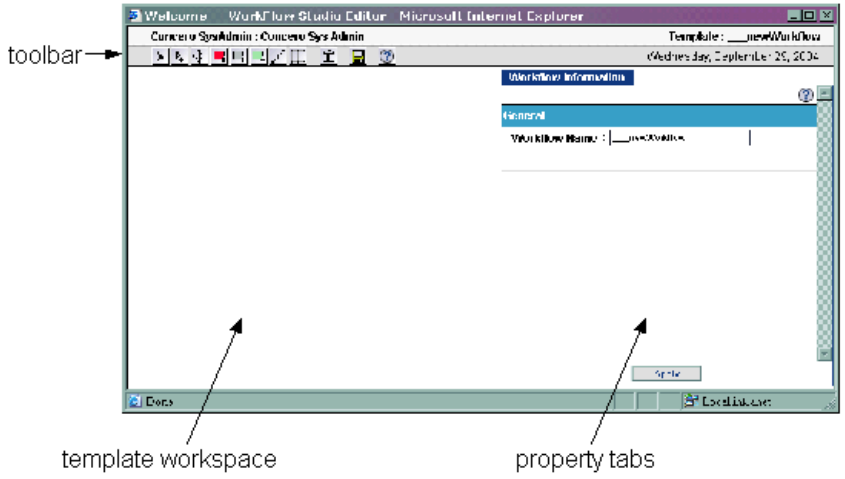
The Workflow Studio interface is very rudimentary. It is implemented in HTML and backed by XML files that define the elements from which you can choose to create a template. The interface is easy to use but it is not necessarily intuitive. This chapter is provided to help you become comfortable using Workflow Studio and its controls.

To launch Workflow Studio, click **Add New Template** on the Workflow Studio home page of the Select Identity client. You can also modify an existing template, such as one of the default templates.



Overview of the Interface

When you click **Add New Template** on the Workflow Studio home page, or when you select an existing template, select **Modify Workflow**, and click **Submit**, the Workflow Studio displays in a new window. The following illustrates the areas of the interface:



You must use the buttons on the **toolbar** to create, select, and move components in the **template workspace**. When you select a component, its properties are displayed on the **property tabs**.

The following table describes the buttons on the toolbar:

Button	Name and Shortcut	Description
	Select (V)	Selects an activity or transaction in the template workspace. If you click this button then click in the template workspace, template properties are displayed.
	Multiple Select (no shortcut)	Selects multiple activities or transactions in the template workspace.

Button	Name and Shortcut	Description
	Move (X)	Moves an activity to the clicked spot in the template workspace. (You cannot use this button to drag and drop an activity.)
	Add End Block (no shortcut)	Adds an end block activity to the clicked location in the template workspace.
	Add Start Block (no shortcut)	Adds a start block activity to the template workspace.
	Add Activity (A)	Adds an activity to the clicked spot in the template workspace.
	Add Transition (C)	Adds a transition between the selected activities.
	Grid On/Off (no shortcut)	Toggles the display of the grid in the template workspace.
	Delete (D)	Deletes the selected activity or transition.
	Save Workflow (no shortcut)	Saves the workflow template.
	Help (no shortcut)	Displays the online help system.

The following graphics represent the building blocks of workflow templates:

	An activity.
	A transition.

The following provides an overview of the steps you must take to create a workflow template. Details about each step are provided in Creating a Workflow Template on page 38:

1. From the home page of WorkFlow Studio, click **Add New Template**. This displays the Workflow Studio, an interface dedicated to creating workflow templates.
2. Specify a name for the template and define its global attributes on the Workflow Information tab (on the right). See Properties and Variables on page 42 for details.
3. Create **activities**. An activity represents a step that occurs in a workflow, such as obtaining approver email addresses or checking the status of a stage. For each activity, you can define properties and actions.

   If you need to group activities, you can create **blocks**. A block is a special group of activities that enables you to define information to be used by the contained activities and to provide block-level reporting.

   See Activities and Blocks on page 50 for details.
4. Create **transitions** between activities. A transition instructs Select Identity how to proceed from one activity to the next. You can set a condition for the transition, to ensure that the transition is not executed until the condition is met. To understand how to create transitions, see Transitions on page 70.
5. Repeat Step 3 through Step 4 as necessary. The creation process for workflow templates is iterative and may require that you rework the logic and flow of the template.
6. Save the template.

To view the status of workflows, see Reporting on page 71 for details.

Default Workflow Templates

Select Identity provides default workflow templates, each of which is an example of a common workflow process. You can assign the default templates to Business Relationships as you become familiar with the system. Edit these templates or use them as they are.

- **SI Provisioning Only**

  This template simply provisions the user. Here are the steps that are executed in instances of this template:

  a. The Provision block provisions the user in the external resource.

  b. If the Provision block is unsuccessful, the workflow transitions into the Retry block, to retry the provisioning after a minute. The template allows for three retries.

  c. If the provisioning fails more than three times, the template logs the failure and the workflow terminates.

  d. If provisioning was successful, the PostProvision block executes the PostProvisionConcero action, which updates the user in Select Identity.

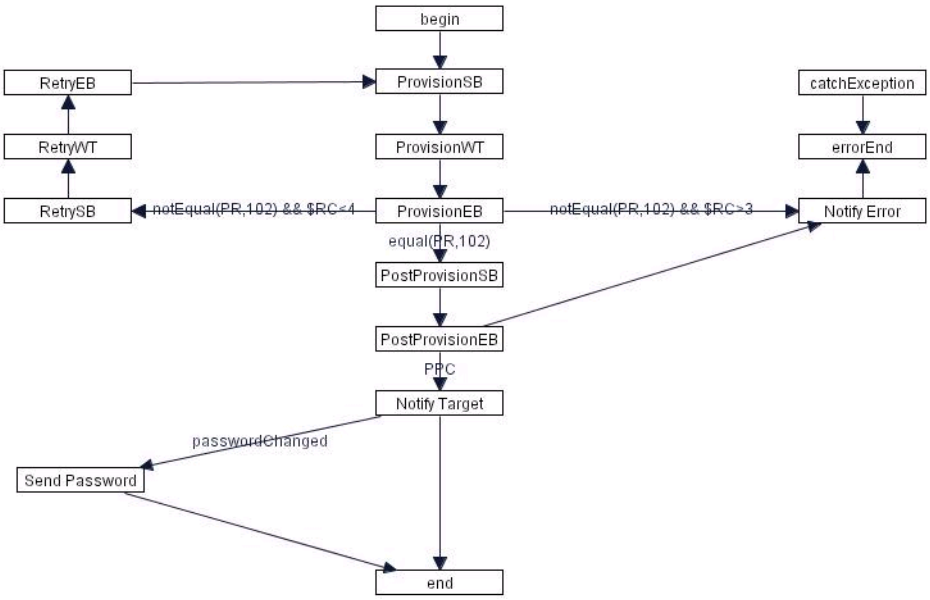
  e. If the PostProvision block fails, the template logs the failure and exits.

  f. If the user was successfully provisioned in Select Identity, the workflow transitions to the Notify Target block, which sends email using the PostAddNotification template. It also determines if the user's password has changed and sets several variables.

  g. If the user's password has changed, the workflow sends email using the NewAccountPassword template.

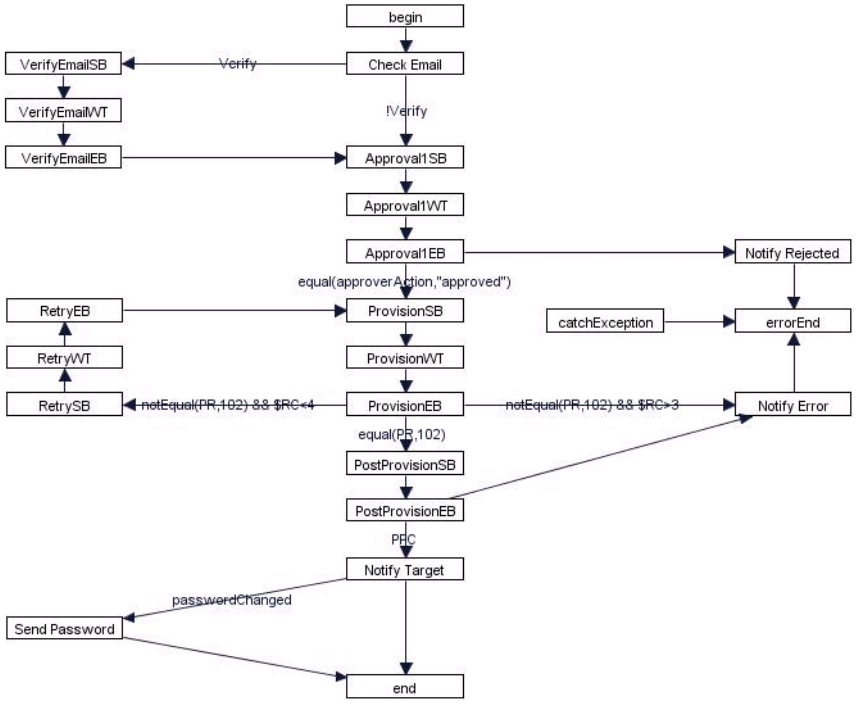
  h. If an exception occurs at any time, the workflow catches the exception and exits.

Here is a snapshot of the SI Provisioning Only template:



- **SI EmailVerifyAndApproval**

  This template verifies the user's email address in Select Identity, requests approval, then provisions the user. Here is a snapshot of the template:



  Here are the steps that are executed:

  a. The CheckEmailVerification action is invoked in the Check Email activity, which sets the a property to false if the email address exists in Select Identity.

  b. If the email address does not exist, the workflow transitions to the VerifyEmail block. This block invokes the saveNotification action, which saves an email notification to a batch table. When the batch process runs, the email is sent. The VerifyEmailEB activity sets the Block Type property, for reporting on the block, and sets the Join Count property. The Join Count property can be used in blocks that require human interaction (such as clicking a link) to keep track of transitions if more than one action must occur before the block transitions from a wait activity.

c The workflow transitions to the Approval block. The GetApproversBySpecifiedRole action is invoked to retrieve the list of approvers for the user. The NotifySelectedApprovers action is invoked to send email to the list of approvers retrieved in the previous activity. The CreateWorkflowTask action is invoked to create an approval task on the Approvals home page for all users in the approver list. This activity is a wait activity, thus the workflow will not transition until the approver accepts or rejects the approval request. Finally, the end block activity sets properties that are passed back to the workflow, including those that govern alert and escalation notification.

d If the approver rejects the request, the Notify Rejected activity invokes the emailNotify action to send mail to an administrator indicating that the account was rejected.

   If the request is approved, the workflow transitions to the Provision block. The ProvisioningTask action is invoked to provision the user in the external system(s) as defined by the Service. The block also sets the Block Type property and invokes the Set variable action to set the failureMessage property.

e If the provision fails, the workflow transitions to the Retry block. The workflow will retry provisioning three times. If it fails after three retries, the workflow transitions to the Notify Error activity, which logs the failure. The workflow then terminates.

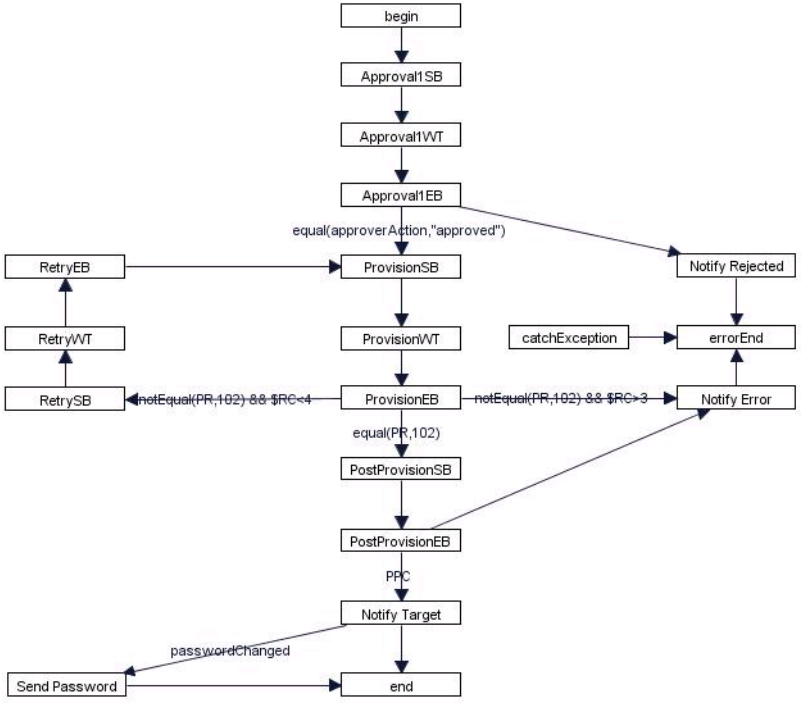
f If the provision succeeds in the external system, the PostProvision block begins. This block provisions the user in Select Identity, sets the Block Type property for reporting, and resets the failureMessage property.

   If post-provisioning succeeds in Select Identity, the Notify Target activity sends email to the user and the Send Password activity sends email to the administrator.

g If the post-provisioning fails, the workflow transitions to the Notify Error activity, which logs the failure. The workflow then exists.

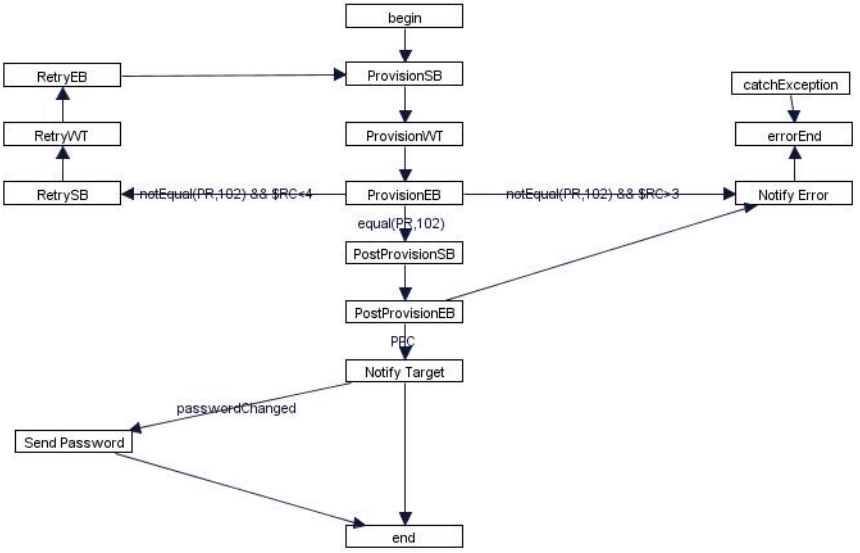
- **SI OneStageApproval**

  This template requests approval then provisions the user. It is identical to SI EmailVerifyAndApproval except that it does not verify the user's email address, nor does it handle errors and exceptions. Here is a snapshot of the template:



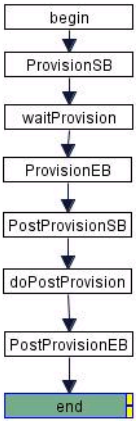
- **SI TwoStageApproval**

  This template is adds another approval stage to the SI OneStageApproval template:



- **ReconciliationDefaultProcess**

  This template reconciles account data with a resource. It is very similar to the SI Provisioning Only template; it provisions users in an external resource. Then, if the user is successfully provisioned, the template executes the ReconciliationPostProvision action, which synchronizes the Select Identity database after reconciliation.

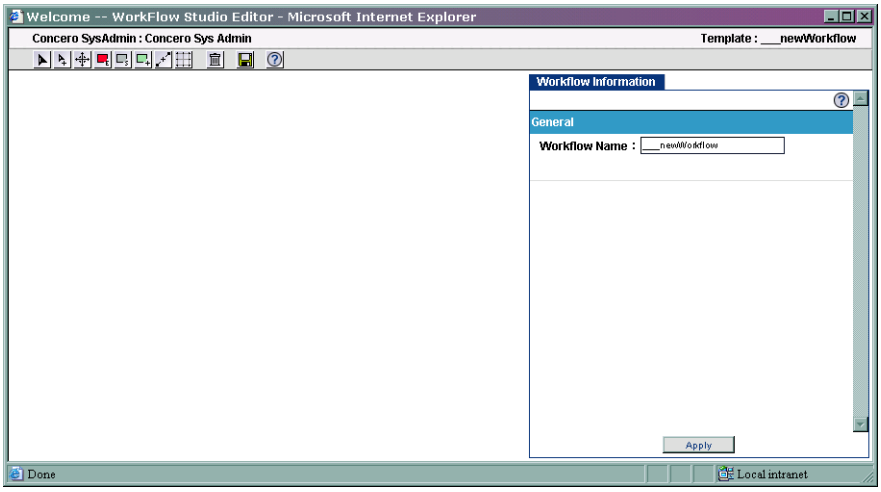


Creating a Provisioning Template

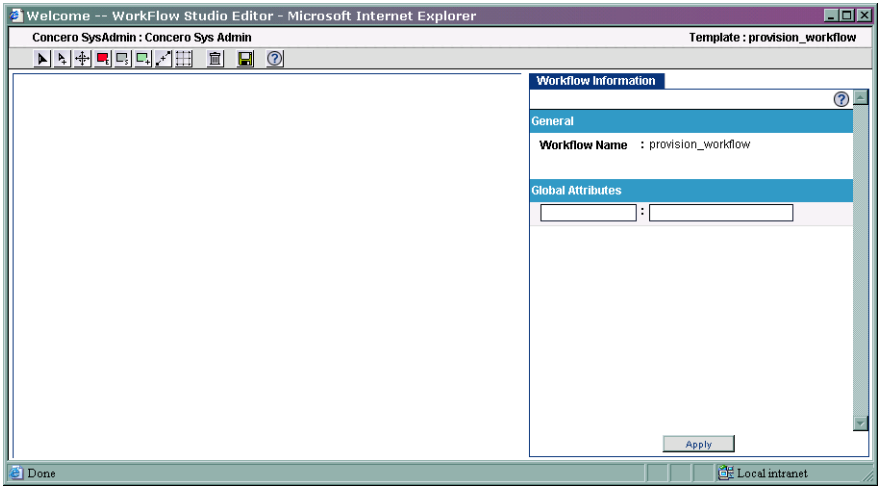
The SI Provisioning Only template provides a workflow for provisioning users. It is a good example of a workflow template, and the steps below walk you through the procedure of creating a similar template. Follow this procedure to practice using the Workflow Studio, to become comfortable with its controls and properties.

As you specify properties for the template, you must click **Apply** to save the information. If you do not click **Apply** after entering data, your changes will be lost.

1 From the Workflow Studio home page on the Select Identity client, click **Add New Template**. This displays the Workflow Studio.

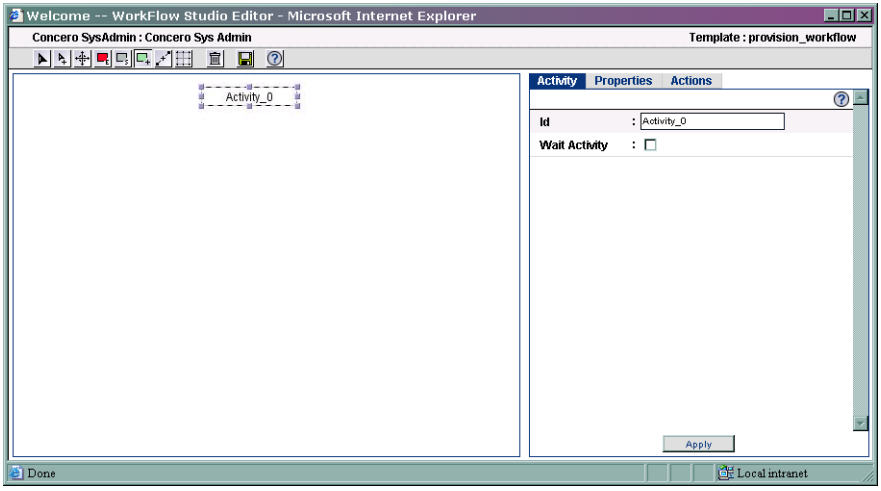


2 Enter **`provision_workflow`** in the Workflow Name field and click **Apply**. The general template properties are displayed on the Workflow Information tab.



3 Create the begin activity:

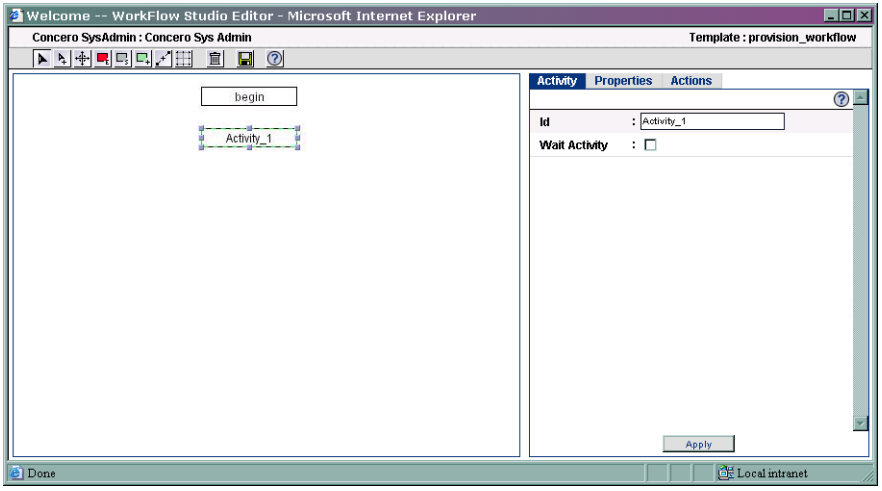
a Click the Add Activity button ( ) then click in the template workspace. An activity is added to the template and the activity's properties are displayed on the property tabs.



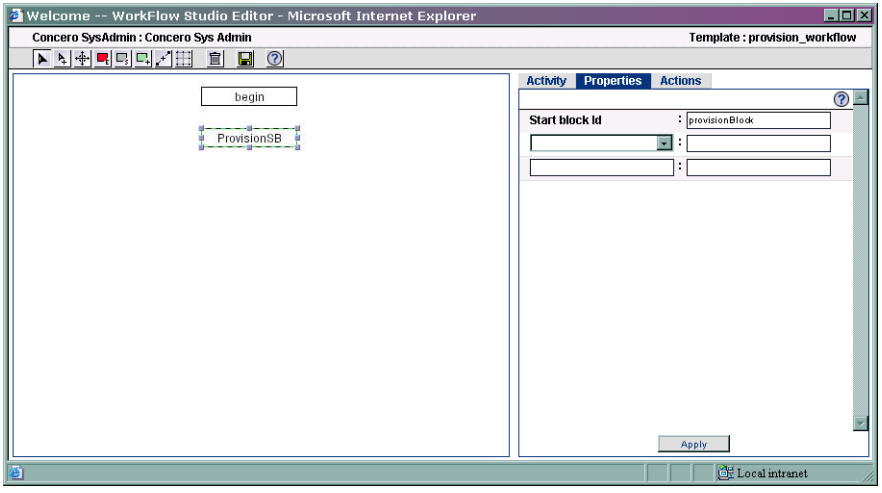
b Rename the activity by entering **`begin`** in the Id field on the Activity tab, then click **Apply**.

4 Create the provisioning block:

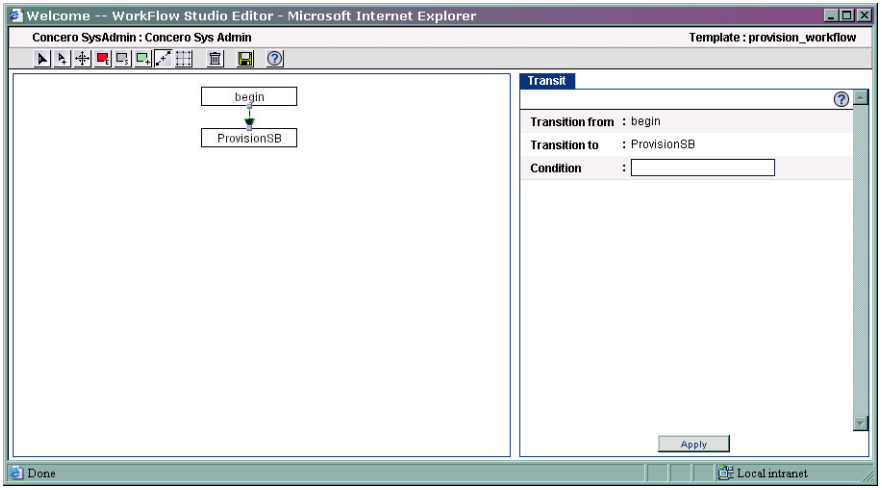
a Click then click below the begin activity in the template workspace.



b Specify properties for this activity. Enter **`ProvisionSB`** in the Id field and click **Apply**. Click the **Properties** tab, which enables you to specify properties for the block. Select **Start block Id** from the drop-down list, enter **`provisionBlock`** in the field on the right, and click **Apply**. This defines the block-level property that begins the block.



c Add a transition from the begin activity to the ProvisionSB activity. Click , click the begin activity, click , and click the ProvisionSB activity. An arrow is drawn between the two activities.



d Add the next activity to the block. Click then click below the ProvisionSB activity in the template workspace. Activity_2 is added to the template.

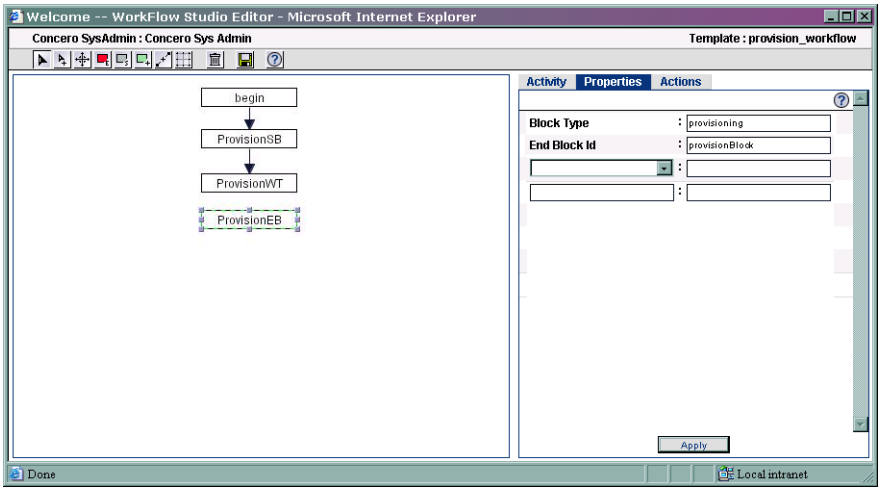
e Define the properties for the activity, which will perform the provisioning actions for the block. Enter **`ProvisionWT`** in the Id field and select the **Wait Activity** check box (this activity must wait for the action to complete before it transitions to the next activity). Click **Apply**.

To define the provisioning action for this activity, click the **Actions** tab then click the **Add** button. Select **Application Invocation** from the Name drop-down list, select **ProvisioningTask** from the Application Name drop-down list and click **Apply**.

f Add a transition from the ProvisionSB activity to the ProvisionWT activity. Click , click the ProvisionSB activity, click , and click the ProvisionWT activity.

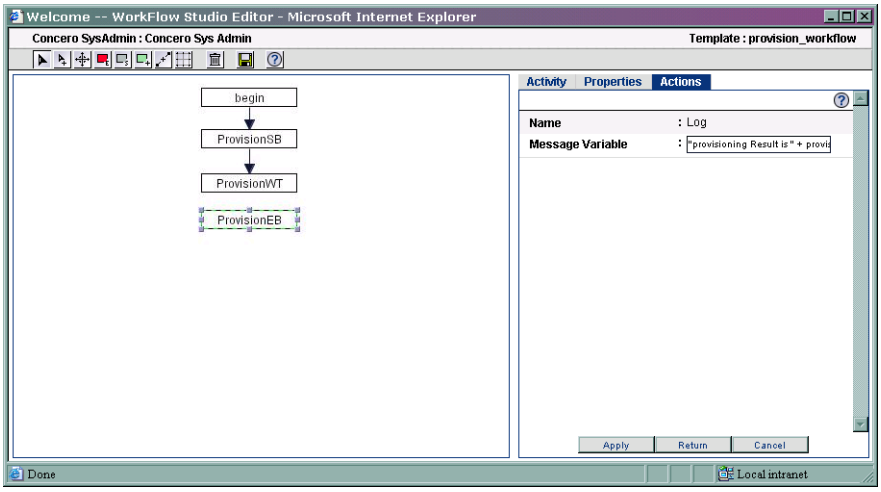
g Add the end block activity to the block. Click then click below the ProvisionWT activity in the template workspace. Activity_3 is added to the template.

h Define the properties of the end block activity. Enter **`ProvisionEB`** in the Id field and click **Apply**. Define a property that will enable you to report on this block later. Click the**Properties** tab, select **Block Type** from the drop-down list, enter **`provisioning`** in the next field, and click **Apply**. You must also specify that this is the end of the block by defining the End Block Id property. Select **End Block Id** from the drop-down list, enter **`provisionBlock`** in the field on the right, and click **Apply**.



Finally, add an action that logs the provisioning result; this action also sets the provisioningResult property. Click **Actions**, click the **Add** button, select **Log Message** from the Name drop-down list, enter

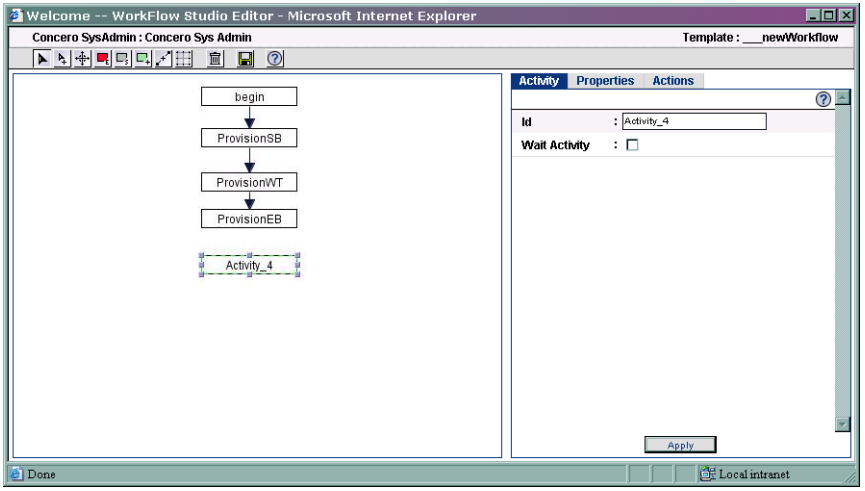
`"provisioning Result is " + provisioningResult` in the Message Variable field, and click **Apply**.



i Add a transition from the ProvisionWT activity to the ProvisionEB activity in the block. Click ▶, click the ProvisionWT activity, click ↗, and click the ProvisionEB activity.

5 Define the post-provisioning block that should be executed if the provisioning block is successful.

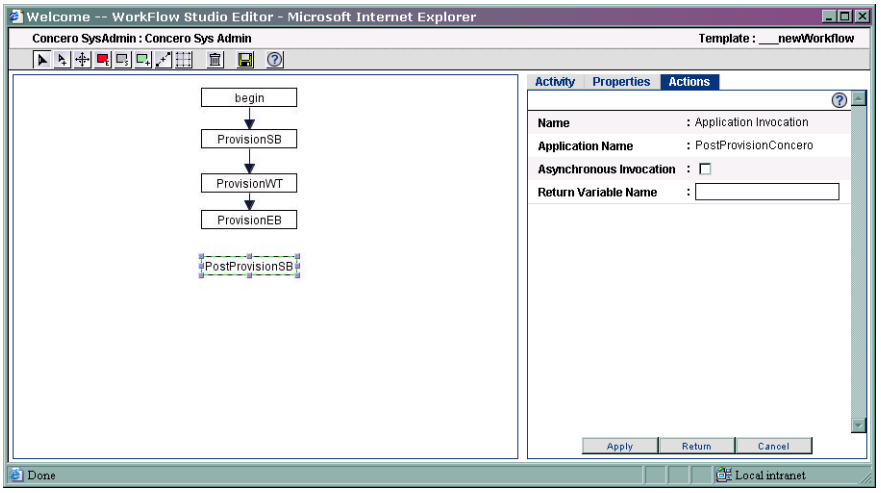
a Click then click below the ProvisionEB activity in the template workspace. Activity_4 is added.



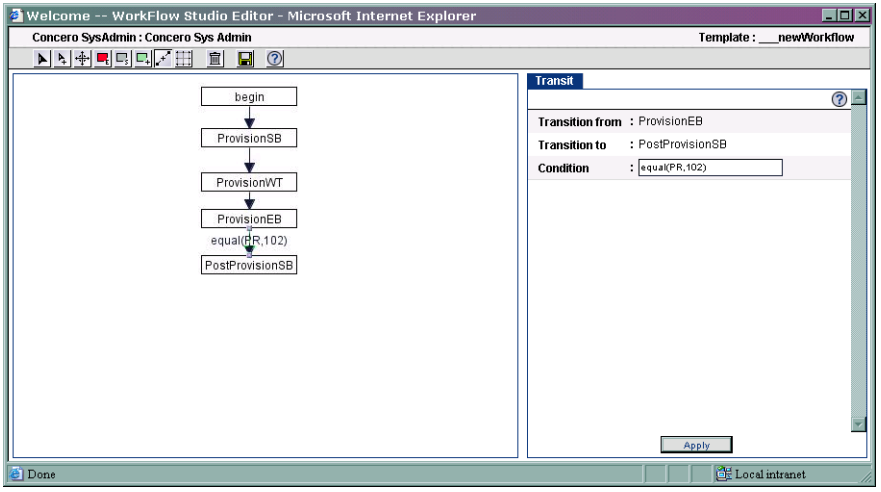
b Specify properties for this activity. Enter **`PostProvisionSB`** in the Id field and click **Apply**. Click the **Properties** tab, select **Start block Id** from the drop-down list, enter **`postProvisionBlock`** in the field on the right, and click **Apply**.

Then, add an action that provisions the user in Select Identity. Click **Actions**, click the **Add** button, select **Application Invocation** from the

Name drop-down list, select **PostProvisionConcero** from Application Name drop-down list, and click **Apply**.



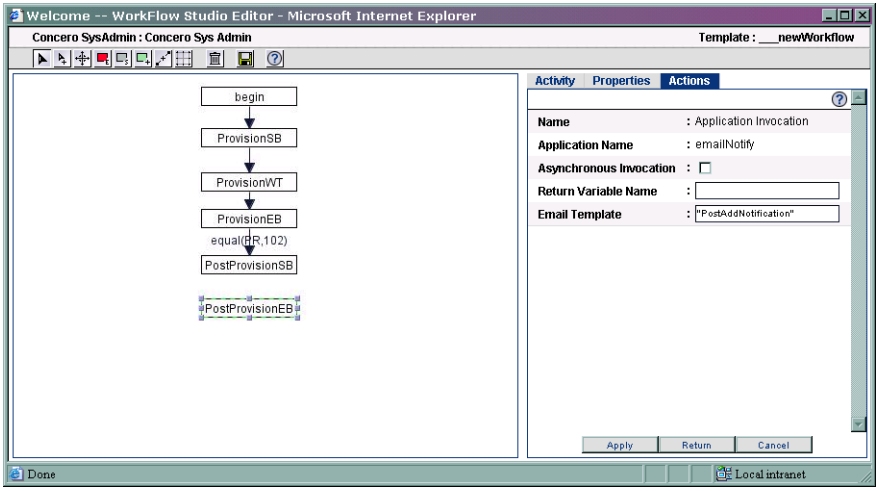
c Add a transition from the ProvisionEB activity to the PostProvisionSB activity. This transition will define the condition that the workflow should not execute the post-provisioning block unless the provisioning block is successful. Click , click the ProvisionEB activity, click , and click the PostProvisionSB activity. In the Condition field, enter **`equal(PR,102)`**, then click **Apply**.



d Add the end activity to the block. Click then click below the PostProvisionSB activity in the template workspace. Activity_5 is added to the template.

e Define the properties of the end activity. Enter **`PostProvisionEB`** in the Id field and click **Apply**. Define a property for reporting purposes. Click **Properties**, select **Block Type** from the drop-down list, enter **`postprovisioning`** in the next field, and click **Apply**. You must also specify that this is the end of the block by defining the End Block Id property. Select **End Block Id** from the drop-down list, enter **`postProvisionBlock`** in the second field on the right, and click **Apply**.

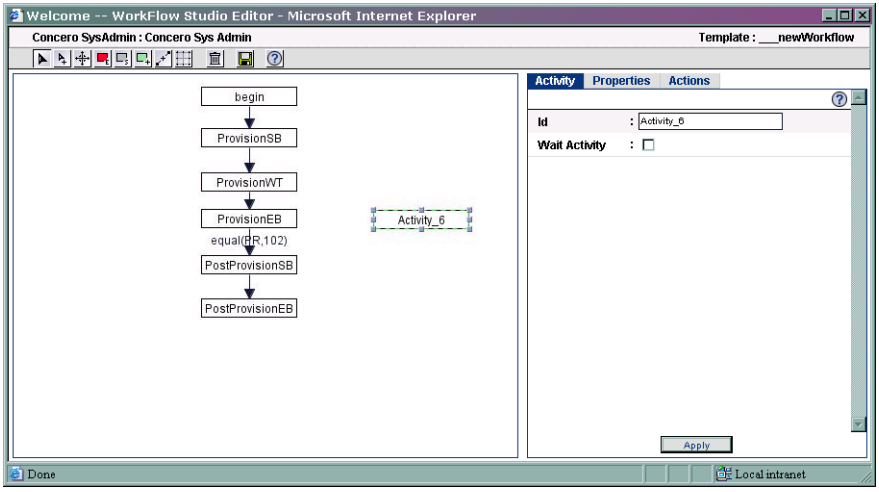
Finally, add an action that sends email to notify an administrator that the provisioning was successful. Click **Actions**, click the **Add** button, select **Application Invocation** from the Name drop-down list, select **emailNotify** from Application Name drop-down list, enter **`"PostAddNotification"`** in the emailTemplate field (this is a default email template), and click **Apply**.



f Add a transition from the PostProvisionSB activity to the PostProvisionEB activity in the block. Click , click the PostProvisionSB activity, click , and click the PostProvisionEB activity.

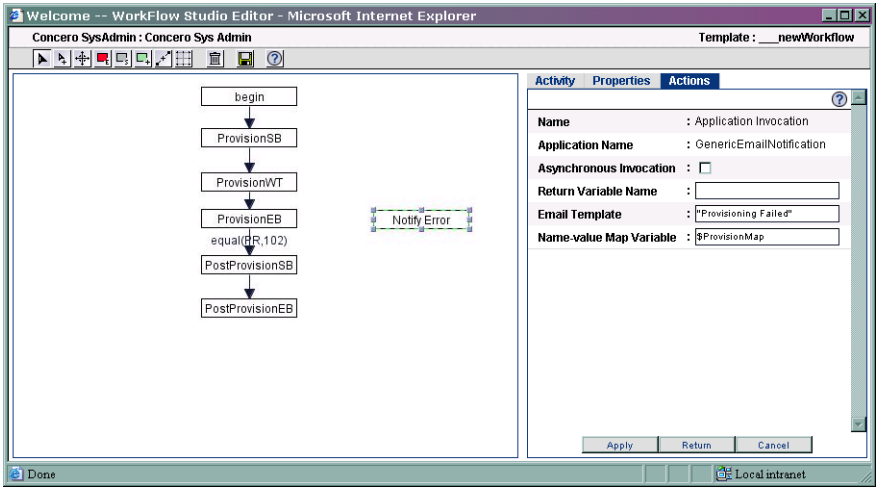
6 Define an activity that will log an error if the provisioning block is unsuccessful.

a Click then click to the right of the ProvisionEB activity. Activity_6 is added.

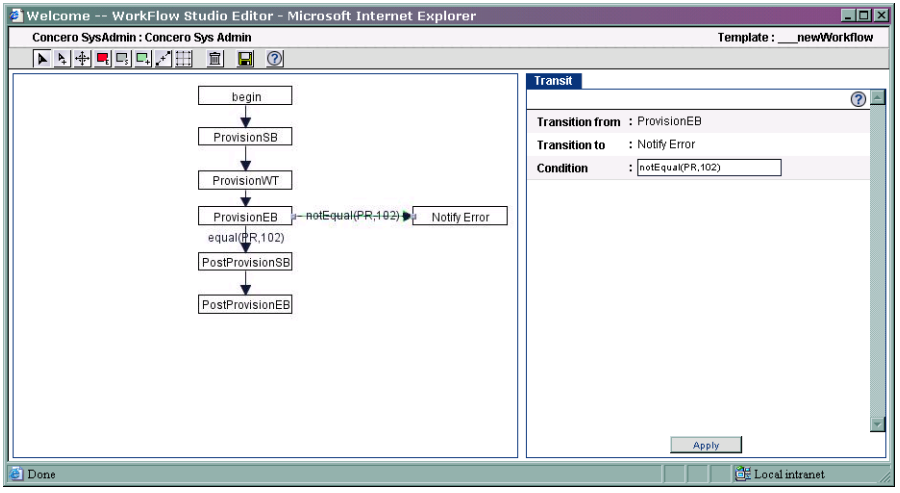


b Specify properties for this activity. Enter **`Notify Error`** in the Id field and click **Apply**. Click **Actions**, click **Add**, select **Application Invocation** from the Name drop-down list, select **GenericEmailNotification** from the Application Name drop-down list,

and enter **"Provisioning Failed"** in the Email Template field, enter **$ProvisionMap** in the Name-value Map Variable field, and click **Apply**.



c Add a transition from the ProvisionEB activity to the Notify Error activity. Click , click the ProvisionEB activity, click , and click the Notify Error activity. In the Condition field on the Transit tab, enter **notEqual(PR,102)**, and click **Apply**.



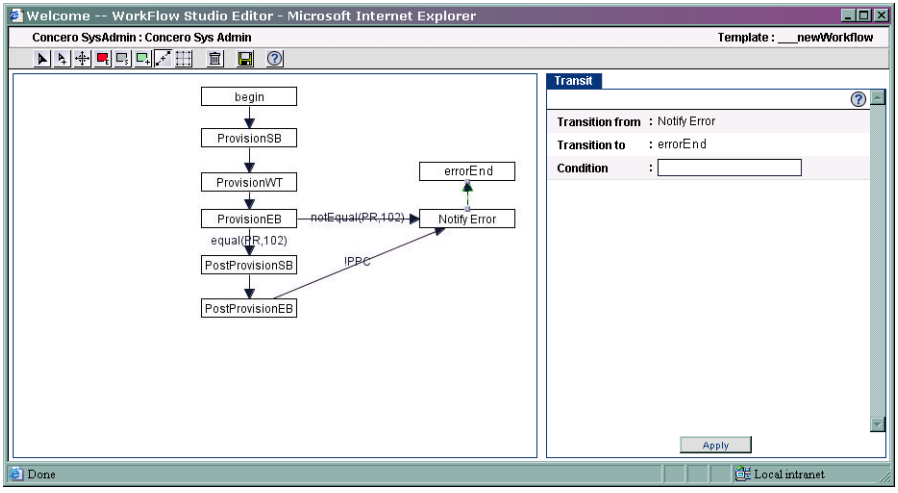
d Add a transition from PostProvisionEB activity to the Notify Error activity, to handle an error in the PostProvision block. Click , click the PostProvisionEB activity, click , and click the Notify Error activity. In the Condition field on the Transit tab, enter **`!PPC`** and click **Apply**. This indicates that the PostProvisionEB activity should transition to Notify Error only if post-provisioning was unsuccessful (the PostProvisionConcero action returns PPC if it is successful).

7 Add an end activity to the template that is used if an error occurred and provisioning could not occur. When the workflow ends with the errorEnd activity, the "End with error" status is displayed on Request Status page of the Select Identity client.

a Click then click above the PostProvisionEB activity. Activity_7 is added.

b Enter **`errorEnd`** in the Id field and click **Apply**.

c Add a transition from Notify Error activity to the errorEnd activity. Click , click the Notify Error activity, click , and click the errorEnd activity.

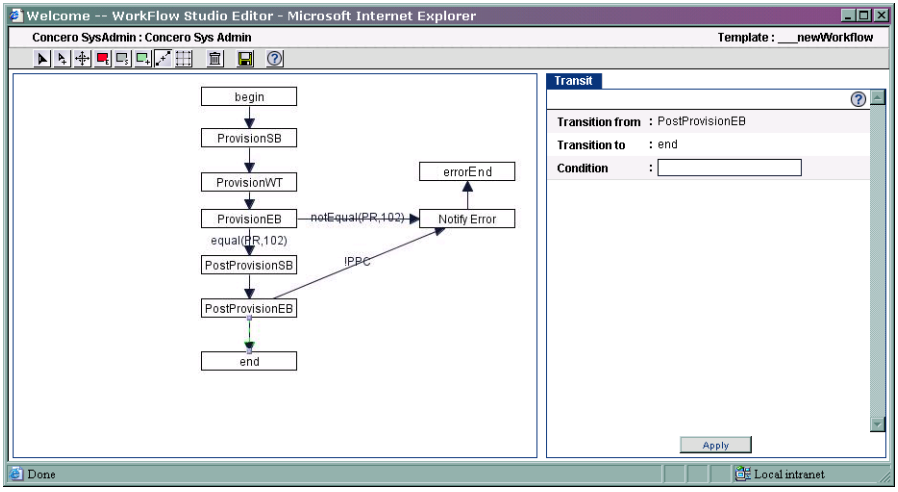


8 Add an end activity to the template that runs if provisioning was successful:

a Click then click below the PostProvisionEB activity. Activity_8 is added.

b Specify properties for this activity. Enter **end** in the Id field and click **Apply**.

c Add a transition from the PostProvisionEB activity to the end activity. Click , click the PostProvisionEB activity, click , and click the end activity.



9 Save the template by clicking .

3

Creating a Workflow Template

Workflow templates provide several features that support robust logic, giving you great flexibility when creating the flow of workflows. When creating workflow templates, consider the following:

- Task logic — Which workflow activities can be handled by simple activities, which should be represented by blocks, and when should you invoke a subworkflow? Blocks provide a way to group activities.
- External applications — Are there external systems or scripts that will perform tasks during the workflow? For example, do you need to provision users in an external system using a connector based on approver feedback? If so, you can register external applications with Select Identity so that you can call on them from Workflow Studio.
- Branching — How should one activity progress to the next? Is there a direct path or is the progression based on some condition?
- Auditing and escalation — What information do you need to track throughout the life of the workflow? You can set temporary properties that track information for a single activity, you can track information in blocks, or you can store information to be used throughout the life of the workflow.

Also, before creating a workflow template, you may need to create dependent information using the Select Identity client, as follows:

- If you plan to request approvals from or notify administrators of changes, you can obtain a list of approvers by specifying a role created on the Admin Roles home page.
- If you intend to send email as part of the workflow, you can reference an email templates created on the Notifications home page.
- If you must perform provisioning actions on external systems, such as to enforce entitlement rules, you can make calls to these systems using external calls registered with Select Identity on the External Calls home page.

Also, keep the following in mind when creating workflow templates:

- When specifying information for an activity, be sure to click **Apply** before loading other pages. If you do not update Studio with your changes, you may lose newly updated properties.
- If you have an existing workflow template open, open a new browser window if you wish to create or modify another template. If you do not open a new window, Workflow Studio will load the template in the current window, overwriting unsaved changes.

It is recommended that you create a dummy workflow template, to practice using the Workflow Studio controls. They can be difficult to use at first. See Getting Started on page 14 for examples you can work through to become familiar with the interface. After it becomes intuitive, create a simple workflow template that you can assign to a request event in a Business Relationship. Then, build upon the simple template or create a template that you can truly use in user provisioning.

Overview of Template Actions

You can use workflow templates to perform an array of tasks in order to provision users. Workflow Studio's flexibility enables you to create templates to process the most complex provisioning actions. In general, though, there is a finite number of actions you need to perform as part of provisioning. The following table lists the tasks you can perform and the template action you must configure to represent the action in Workflow Studio.

Tasks	Actions in Workflow Studio
Variables — Retrieving and setting variables, to pass data to activities in a workflow and to external processes	• Add Item to List on page 67 • Add Item to Map on page 68 • XPath on page 69 • Set Variable on page 66
Approvers — Obtaining and notifying lists of approvers	• Application Invocation → GetApproversBySpecifiedRole on page 57 • Application Invocation → NotifySelectedApprovers on page 56 • Application Invocation → CreateWorkflowTask on page 58 • Application Invocation → ApproversExternalCall on page 65
Email — Obtaining email addresses and sending email	• Application Invocation → emailNotify on page 60 • Application Invocation → GenericEmailNotification on page 61 • Application Invocation → CheckEmailVerification on page 54 • Send Email on page 68

Tasks	Actions in Workflow Studio
Notifications — Sending notifications using notification policy and templates	• Application Invocation → CheckEmailVerification on page 54 • Application Invocation → emailNotify on page 60 • Application Invocation → GenericEmailNotification on page 61 • Application Invocation → NotifySelectedApprovers on page 56 • Application Invocation → saveNotification on page 55
Provisioning — Provisioning users in Select Identity and external resources, and invoking scripts and external calls	• Application Invocation → ProvisioningTask on page 59 • Application Invocation → PostProvisionConcero on page 62 • Application Invocation → ReconciliationPostProvisioning on page 63 • Application Invocation → ExternalCall on page 64 • Application Invocation → ApproversExternalCall on page 65 • Run Script on page 66 • Call Subworkflow on page 67
Logging and errors — Logging messages to the Select Identity log handler and handling errors	• Log Message on page 66 • Throw Exception on page 66 • Recover from Last Error on page 69

Properties and Variables

When creating a workflow template, you can define properties. You can also use variables to pass data to and from the workflow engine. Both properties and variables can be referenced in the report template to display information on the Request Status page on the Select Identity client. Workflow variables exist in the scope of a workflow instance; properties exist in the scope of workflow template.

Using Properties

A property is simply a name-value pair and the value is a string. Properties define constant data when the template is created; property values do not change at runtime. A global property is shared by all the activities within a workflow instance. The first time you set a property, you initialize its type. See Activities and Blocks on page 50 for details.

The specified properties can be read by external applications using the Workflow API provided by Select Identity. They can also be referenced by a report template to show relevant information in a status report.

Some property names are defined by the workflow engine. Use these properties when defining activities and blocks. Assign values to these properties when you create the workflow template to instruct the workflow how to operate. For example, you could assign a value of 3 to the Join Count property. The workflow would then wait for three of approvers to join the workflow before it exits the wait defined in the approval block.

The following table describes the system-defined properties:

Property	Where	Description
reportType	Workflow Information tab (template property)	Specifies the name of the report template to use. The engine will use the referenced report template to render the workflow instance data. See Report Templates on page 74 for more information.
Start block Id	Start block activity	Sets the block ID in a start block activity. This value must be unique within the template and must match the value of End Block Id. The workflow engine uses this ID to identify a block. Any activities between the start and end block activities are included in the same block.
End Block Id	End block activity	Sets the block ID in the end block activity. The assigned value must match that of Start block Id.

Property	Where	Description
Block Type	End block activity	Specifies the block type, which is used by the report engine when rendering information in the status report. All blocks of the same type in a workflow template are rendered according to the report template. If Block Type is not specified, Select Identity uses the default report template to render information in reports. The following block types are defined in the default report template: • For blocks that verify email, specify **`emailVerify`** to report on this block. • For approval blocks, specify **`approval`** to report on this block. • For the provisioning block, specify **`provisioning`** to report on this block. • For the post-provisioning block, specify **`postprovisioning`** to report on this block. • Specify **`ReconciliationPostProvision`** to report on reconciliation blocks.
Join Count	End block activity	Specifies the total number of anticipated joining transitions. When this number is equal to the joined count, the end block or AND join activity exits.
Role Name	End block activity	Specifies the role name (defined in Select Identity) that is used by the GetApproversBySpecifiedRole action defined in the same activity.
Escalate To	End block activity	Specifies the email address of the user to whom the escalation message is sent. Specify this property in an approval block.

Property	Where	Description
Escalation Handler	End block activity	Specifies how the escalation is delivered. Specify a string (but do not quote the string). Currently, you can only specify **`email`** as the value of this property. Specify this property in an approval block.
Escalation Timeout	End block activity	Sends a reminder notification to the address specified by escalateTo when the workflow instance times out. Specify the timeout value in the following format: [*n* day[s]] [*x* hour[s] | hr[s]] [*y* minute[s] | min[s]] [*z* [second[s] | sec[s]] where [] indicates an optional parameter and | indicates OR. If you simply specify an integer, the workflow engine interprets the time as seconds. The following are examples: 7 days 1 day 12 hrs 30 mins 45 seconds 3600 Set this property in an approval block.
Escalation Repeat Count	End block activity	Directs the workflow engine to repeat the escalation notification. Specify how many times to repeat the escalation (integer). Specify this property in an approval block.
Alert Name	End block activity	Specifies the name of the person who is alerted. Specify this property in an approval block.
Alert Handler	End block activity	Specifies how the alert is delivered. Specify a string (but do not quote the string). Currently, you can only specify **`email`** as the value of this property. Specify this property in an approval block.

Property	Where	Description
Alert Timeout	End block activity	Sends an alert to the address specified by Alert Name when the workflow instance times out. Specify the timeout value in the following format: [*n* day[s]] [*x* hour[s] | hr[s]] [*y* minute[s] | min[s]] [*z* [second[s] | sec[s]] where [] indicates an optional parameter and | indicates OR. If you simply specify an integer, the workflow engine interprets the time as seconds. The following are examples: 1 day 5 hrs 20 mins 12 hours 2400 Set this property in an approval block.
Alert Repeat Count	Block	Directs the workflow engine to repeat the alert. Specify how many times to repeat the alert (integer). Specify this property in an approval block.
Reminder Handler	End block activity	Specifies how the reminder is delivered. Specify a string (but do not quote the string). Currently, you can only specify **`email`** as the value of this property. Specify this property in an approval block.

Property	Where	Description
Reminder Timeout	End block activity	Sends a reminder notification to the address specified by Reminder Handler when the workflow instance times out. Specify the timeout value in the following format: [*n* day[s]] [*x* hour[s] | hr[s]] [*y* minute[s] | min[s]] [*z* [second[s] | sec[s]] where [] indicates an optional parameter and | indicates OR. If you simply specify an integer, the workflow engine interprets the time as seconds. The following are examples: 2 days 5 hours 30 minutes 1 day 3600 Set this property in an approval block.
Reminder Repeat Count	End block activity	Directs the workflow engine to repeat the reminder notification. Specify an integer representing the number of times to repeat the reminder. Specify this property in an approval block.

Property	Where	Description
Timeout At	End block activity	Transitions the block back to the workflow when the workflow instance times out. Specify the timeout value in the following format: [*n* day[s]] [*x* hour[s] | hr[s]] [*y* minute[s] | min[s]] [*z* [second[s] | sec[s]] where [] indicates an optional parameter and | indicates OR. If you simply specify an integer, the workflow engine interprets the time as seconds. The following are examples: 1 day 12 hrs 30 minutes 1 hour 60
Timeout After	End block activity	Sets the timeout value, in hours (integer), for the block. The block transitions back to the workflow when the waiting time exceeds this timeout.

Using Variables

A variable defines dynamic data that is set and changed while the workflow instance is running. Like properties, a variable is a name-value pair. Variables can be created or changed at run-time in a workflow instance through actions, a Workflow API call, or returned by an application invocation. To create or change a variable in a template, use the Set Variable action.

Assign a qualified variable string to the name of the variable. In addition, keep these guidelines in mind when naming a variable:

- The name cannot include spaces.
- The first character of the name cannot be a number
- The only special characters allowed in variable names are underscores (_) and dollar signs ($). Otherwise, use only alpha-numeric characters.

You can assign any of the following to the value of a variable:

- A constant (string, integer, and so on)
  If you assign a string constant, quote the string by surrounding it with double quotes. Here is an example: `Var1 = "this is a string."` When Var1 is evaluated, it resolves to `This is a string`.

  If you assign an integer to a variable, do not quote the value, as in this example: `Var2 = 4`

- Another variable
  Simply assign the name of the variable as the value of your variable. If a variable called Data is assigned the string "This is data", then assigning the Data variable to Var2 (`Var2 = Data`) resolves as follows: `Var2 = "This is data"`

- An expression
  An expression can contain variables and constant values. For example, if a workflow variable called Status is assigned the value "OK", the following expression will resolve as `The status is OK`:

  ```
  "The status is" + Status
  ```

  This illustrates an expression that appends the value of a workflow variable to a string.

  A set of workflow macros are defined for use in expressions. For example, in a transition condition, you can use the equal() macro to compare two objects, as in this example: equal(approved,"approved"). The workflow engine will evaluate it and make a decision based on the returned result. Here are the macros defined for use in workflow templates:

  equal(a, b) — if a equals b, the macro returns true, otherwise false

  notEqual(a,b) — if a is not equal to b, the macro returns true, otherwise false

  getListItem(listVariableName, index) — returns a list item

  getMapItem(mapVariableName, name) — returns a map item

Note that you cannot change the variables type after the variable is initialized.

Workflow Studio supports the following levels of variables:

- Workflow
  At the workflow level, variables are categorized as follows:
  - Persisted — Variable names of persisted variables begin with **$** and are stored in the database, even when a workflow instance is passivated. You can access these variables at any time once the workflow instance is created.
  - Non-persisted — This type of variable is temporary and available in the workflow until a wait activity begins (the workflow is passivated) or until the workflow completes (the workflow terminates).

  You can reference workflow variables in workflow templates, report templates, and in Workflow API calls.
- Block
  Block-level variables are persistent and identified by a variable name and block ID. Use the Set Variable actions to move data between block-level and workflow-level variables.
- Activity
  These variables are temporary and available within an activity only. For example, you may need to get a manager's email address. You can define an activity-level variable to store the email address.

Activities and Blocks

Activities define the tasks that must occur in the workflow. When assigning a name to an activity, ensure that it is unique; activity names (IDs) must be unique in a workflow template. The workflow engine supports the following activity types:

- begin — If you intend to assign the workflow template to a request event, you must create a begin activity as the first activity in the template. Assign **`begin`** (using all lowercase letters) as the name of the activity. You must create a begin activity as the first in the template. The workflow engine will throw an exception if it cannot determine where to begin.

  If the template will not be referenced by a request event, creating a begin activity is not necessary. For example, a subworkflow invoked by another workflow does not require a begin activity. The parent workflow can

explicitly specify the starting activity in the subworkflow (using the activity ID variable). Likewise, workflow templates can be invoked by external applications using the Workflow API. If the workflow is invoked by the API, the begin activity is not necessary and the API can specify the first activity with which to start.

The workflow engine will throw an exception if it cannot determine where to begin.

- Wait — Temporarily pauses a workflow until resumed by a callback. To create a wait activity, select the Wait Activity check box in the activity.
- Start and End Block — The entry and exit activities for a block. Assign the Start block Id property to the start block activity, and assign the End Block Id property to the end block activity. The values of these properties must match.
- Exception — An activity that handles errors that occur when the workflow instance is executed. Designate an activity for exception handling by assigning **`catchException`** as the name of the activity.
- errorEnd — You can create an activity named **`errorEnd`** that terminates the workflow if an error occurs. When the workflow ends with the errorEnd activity, the “End with error” status is displayed on Request Status page of the Select Identity client. If the workflow ends with any other activity, the Request Status page displays “End” status.

Creating Blocks

In addition to individual activities, you can create groups of activities that comprise a block. A block is a group of activities that is bounded by start and end block activities. A block has two purposes:

- To define information to be used by a subset of activities (block-level properties)
- To provide block-level reporting

When creating a block, you must following these guidelines:

- Create the start block activity and assign the Start block Id property to it.
- Add all activities and transitions beneath the start block activity. The start block activity can perform actions, though you may want to perform actions after the start block activity for simplicity.

- Create an end block activity and assign the End Block Id property to it. The End Block Id property *must* match that of the Start block Id property. Also, an end block activity cannot be a wait activity.
- The end block activity should assign the result of the block logic and should set any properties that must be passed back to the workflow. It can also log results to the log handler. Other than this, the end block activity should not perform other actions.
- Any activity within the block cannot jump to an activity outside of the block without going through the end block activity. Likewise, none of activities can jump into the block without going through the start block activity.

When assigning actions to blocks, you can choose any of the available actions (see page 53).

Setting Properties on the Activity Page

You can set the following on the Activity tab for activities:

Property	Description
Id	The name or ID of the activity.
Wait Activity check box	Defines this activity as a wait activity. Only global variables that are persisted exist after a wait activity.

Property	Description
Join Type	If more than one incoming transition is defined for the activity, this drop-down list is displayed on the Activity tab for the selected activity. It enables you to specify that the activity executes if all of the incoming transitions are received (AND) or if the activity executes after only one transition condition is made (XOR). Join type transitions are usually paired with split type transitions, and XOR is the default join type. Also, if no condition (join or split) is specified for a transition, the transition always occurs.
Split Type	If more than one outgoing transition is defined for the activity, this drop-down list is displayed for the selected activity. It enables you to specify that subsequent activities execute if all transition conditions are met (AND) or if only one transition condition is met (XOR). XOR is the default split type.

Defining Actions

Activities contain actions to define the steps in the tasks. Actions actually do the work in the workflow.

Actions are executed in the order in which they are added to the activity.

The workflow engine provides a set of actions that you can assign and configure for activities. These engine-defined actions provide common workflow data and control operations, such as the Set Variable action of the Log Message action. Also provided are application-specific actions (specific to Select Identity).

The following sections describe the actions that are preregistered with Workflow Studio to enable you to perform general and application-specific tasks within an activity. The sections that are provided here are named according to each action you can choose on the Actions tab in Workflow Studio.

Application Invocation

The Application invocation action enables you to call an external application. Select Identity provides many applications that you can use, as described in this section.

CheckEmailVerification

This application determines whether an email address exists in Select Identity. The CheckEmailVerification service returns a boolean indicating the result. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify the name of a variable to store the verification result (a boolean) if you wish to catch the status. It returns true if the user requires verification of the email address, and false if not. Example: **`emailVerification`**

saveNotification

The saveNotification application enables you to create a notification for a user indicating that he or she must verify the specified email address. This application must be invoked from a wait activity. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Select this check box for this service.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. For this service, leave this field blank.

NotifySelectedApprovers

This application notifies approvers of a pending workflow task using the notification mapping created in the Service's Business Relationship. The approvers are those in the role retrieved by GetApproversBySpecifiedRole (page 57). Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify the name of a variable that can store the status (boolean) of the application if you wish to store the status. It returns true if all of the approvers are sent email notifications, and false if a problem occurs sending the email.
Notification Action field	The name of the notification mapping in the service. Example: **`"Approve"`**
Approver List Variable field	The list of approvers. You can specify a variable. Example: **`apprvList`**

GetApproversBySpecifiedRole

The GetApproversBySpecifiedRole application enables you to retrieve a list of approvers based on their role. The role must exist within Select Identity. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box for this service.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. For this service, specify the list of approvers who are assigned the specified role. Example: **`apprvList`**

In addition, you must set the Role Name property for this application. Set this variable in the end block activity. Click **Properties** and enter a property that represents the role. Assign a string to the property; you do not need to quote the value.

Example: **`Role Name`** : **`Workflow Approver`**

CreateWorkflowTask

This application creates a task for all approvers in the specified list on the Approvals home page (in the Select Identity client). Call this application in wait activities only. The approver list is populated by the GetApproversBySpecifiedRole action; see page 57 for details. Set the following properties for this service:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. For CreateWorkflowTask, specify a variable name that can store the return value, which indicates the status (a boolean) of the application invocation, if you wish to store the status of this application. It returns true if all of the relevant information regarding the workflow approval task is persisted in the database successfully, and false if not. Example: **`wfTask`**
Approver List Variable field	The list of approvers, which was returned by the GetApproversBySpecifiedRole action. Specify a variable, such as **`apprvList`**.

ProvisioningTask

This application enables you to provision users on target resources, which are defined by the Service. If provisioning fails on a resource, provisioning on previous resources is rolled back. You can call this application from within a wait activity only.

Set the following properties for this service:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Select this check box for this service.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, leave the field blank.

This application creates a workflow property called **provisioningResult**, which indicates the status of the ProvisionTask operation. You can use in transition conditions and other activities. The following status codes are returned:

- 0 — pending
- 101 — in progress
- 102 — success
- 103 — failure
- 104 — waiting for response

This operation also creates a variable called **errorCode**, which indicates the error information for this operation:

- 0 — no error
- 1001 — no resource(s) to provision

emailNotify

This application sends an email notification to the user being provisioned, and it uses a notification template to send the email. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. If you wish to store the return value that indicates the status (a boolean) of the application invocation, specify a variable name. It returns true if a notification was sent successfully, and false if the notification could not be sent.
Approver Comment field	The contents of the email that is passed to the notification template. If you set this property to **`$WF_APPROVER_COMMENT`**, the email will include all approver comments.
Email Template field	The name of the Select Identity email template to use. Quote (with double quotes) this string.

This application creates a workflow property called **$ApproverComments**, which stores each approver's comments, the data and time of the comment, and the name of the approver.

GenericEmailNotification

This application is used to notify anyone other than the requestor or user being provisioned. For example, you could send email to the user's manager who is not a user provisioned in Select Identity. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify a variable name to store the status (a boolean) if you wish to store the status of this operation. It returns true if the notification was sent to the email target, and false if an exception occurred in the process.
Email Template field	The notification template to use when sending the email. Specify the name of the template in quotes.
Name-value Map Variable field	A map object variable that contains a list of name-value pairs to be used to substitute matched parameters in the email template. Create a map by calling the Add Item to Map action (see page 68) before this action.

PostProvisionConcero

This application synchronizes the Select Identity database with the users that are provisioned (after all provisioning completes successfully). Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify a variable name to store the status (a boolean) if you wish to store the status of this operation. It returns true if post provisioning succeeds, and false if it fails.

ReconciliationPostProvisioning

This application synchronizes the Select Identity database after reconciliation. This application handles all updates done during reconciliation, instead of instantiating a workflow instance for each update. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify a variable name to store the status (a boolean) if you wish to store the status of this operation. It returns true if reconciliation provisioning succeeds, and false if it fails.

ExternalCall

This application makes a call to an external system during workflow approval. The external call must be developed as described in the *HP Openview Select Identity External Call Developer Guide*, which is available on the Select Identity CD in the `docs/api_help/external_calls` directory. The external call must be registered in Select Identity before you can use this service. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify a variable name to store the status (a boolean) if you wish to store the status of this operation. It returns 1 if the external call is successful and 2 if it fails.
External Call Name field	The name of the external call. You can specify name of an Approver Selection or Workflow external call as registered on the External Calls page of the Select Identity client. You must quote this name. Example: **`"checkDB"`**

ApproversExternalCall

This application queries an external system for a list of users who can approve provisioning requests during a workflow. The external call must be developed as described in the *HP Openview Select Identity External Call Developer Guide*, which is available on the Select Identity CD in the `docs/api_help/external_calls` directory. The external call must be registered in Select Identity before you can use this service. Set the following properties for this application:

Property	Description
Asynchronous Invocation check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The application completes. • An exception is thrown in the main workflow. Do not select this check box if you wish to store the return value for this application.
Return Variable Name field	The name of a workflow variable that holds the returned value of this application invocation. Only applications that run in the synchronous invocation mode can return a value. In this case, specify a variable name to store the status (a boolean) if you wish to store the status of this operation. It returns 1 if the external call is successful and 2 if it fails.
External Call Name field	The name of the external call. You can specify name of an Approver Selection or Workflow external call as registered on the External Calls page of the Select Identity client. You must quote this name. Example: **`"checkDB"`**
Default Approver field	A list of Select Identity user names (approvers); if more than one name is specified, separate them with commas. If no approvers are found in the external source, this list of approvers is returned.

Set Variable

This action assigns a value to a workflow variable. If the variable does not exist, this creates the variable and initializes its type. The following properties are provided for this action:

Property	Description
Target Variable Name field	The name of the variable. Do not quote this name.
Source Variable field	The value to assign to the target variable. Specify a string, variable, or expression. Example: `"Mark" + " " + IName`

Log Message

This action logs information to the console window or to a log file, as defined by the logging configuration for the Select Identity server. Refer to the *HP OpenView Select Identity Installation Guide* for configuration details.

For this action, you must set the **Message Variable** property. Specify the value to be logged; you can specify a variable, string, or expression.

Example: `"Current Name =" + nameVar`

Throw Exception

This action throws an exception. For this action, you must set the **Exception Message Variable** property. You can specify a string, variable, or expression. Exceptions are handled by the catchException activity (see page 51 for more information).

Example: `"Error:" + fullName + " not found"`

Run Script

This action is for advanced use only, and it enables you to run a script, such as to manipulate complex business logic for testing purposes. The script must be written in Beanshell (**http://www.beanshell.org**).

For this action, you must set the **Script Variable** property, which specifies the name of a variable that contains the script code.

Call Subworkflow

This action calls another workflow template. The following properties are provided for this action:

Property	Description
Template ID Variable field	The subworkflow's template name.
Asynchronous check box	The invocation mode. For synchronous invocation, the workflow is blocked until one of the following occurs: • The end of subworkflow activity is reached. • A wait activity of the subworkflow is reached. • An exception is thrown in the subworkflow. Do not select this check box.
Activity ID Variable field	The name of the activity where the subworkflow should begin.
Child Callback check box	If selected, the subworkflow will resume the parent workflow when the subworkflow completes.

Add Item to List

This action enables you to create a list for passing multiple variables in a single workflow list variable. (A list is a collection of variables.) You must call this action for each variable you wish to add to the list.

Property	Description
New Collection check box	If you are creating a new list, select this check box.
List Variable Name field	The name of the variable to create or to which you are adding a variable. Do not quote this name.
Element Value Variable field	The value of the list element. Quote the value if this is a string.

Add Item to Map

This action enables you to create a map for passing multiple variables in a single workflow map variable. (A map is a collection of name-value pairs where the value is a workflow variable.) You must create an Add Item to Map action for each variable you need to set in the map.

Property	Description
New Collection field	If you are creating a new map, select this check box.
Map Variable Name field	The name of the map to create or to which you are adding a variable. Do not quote this name.
Element Name Variable field	The name of the variable to set.
Element Value Variable field	The value of the variable. Quote the value if this is a string.

Send Email

This action sends email to anyone without using a notification template. For example, use this action to notify an administrator if an exception occurs.

Property	Description
To Variable field	The email address where the workflow will send the message. You can specify a string, which must be quoted, or the name of another variable.
From Variable field	The sender's email address. You can specify a string, which must be quoted, or the name of another variable.
CC Variable field	The email address of anyone you wish to copy on the message. You can specify a string, which must be quoted, or the name of another variable.
Subject Variable field	The subject of the email. You can specify a string, which must be quoted, or the name of another variable.
Content Variable field	The contents of the email message. You can specify a string, which must be quoted, or the name of another variable.

Recover from Last Error

If an exception is thrown, this action directs the workflow back to the activity where the exception occurred. Using this action implies that you are including logic to fix the problem that caused the exception.

You must specify the **Exception ID Variable** property. Set this property to the name of the activity in the workflow template that threw the exception.

This action is typically used with the catchException activity, which is started when an exception is thrown in the course of workflow instance execution. The catchException can then be followed by a wait activity that forwards an alert message to administrators who can then take action to fix the error. If the The alert message contains the exception context (as specified by the Exception ID Variable property) . After the error is fixed, the administrator can call back to the workflow with the previously received exception ID to resume the workflow. The workflow can then use the exception ID to transition to an activity that invokes the Recover From Last Error action, which resumes the activity that threw the exception.

XPath

This action enables you to search for a string in an XML variable. You can set a variable that contains multiple values.

Property	Description
XML Variable Name field	The name of the <Store> element for which to search.
XPath Variable field	The search path in the following format: ***store/book*[name="*string*"]**.
Result Variable Name field	The variable to contain the returned data.
Results List check box	Select this check box to return a complete list of the strings found.

Transitions

Transitions enable you to connect activities and provide a path for progressing through the workflow. You can also define that certain conditions are met before the workflow progresses from one activity to the next. The Workflow Studio enables you to create transitions as follows:

- You can create a simple transition to connect one activity to the next.
- You can create multiple outgoing transitions. This is called a **split** transition. To indicate how outgoing transitions behave, specify the transition type in the activity that splits into multiple transitions.
  - — Select the XOR split type for one of the outgoing transitions only. When any one of the transitions' conditions are met, the activity transitions to the next and all other transitions are ignored. If none of the conditions are satisfied, the transition that does not specify a condition perform the transaction.
  - — Use the AND split type to transition from one activity to others. If a condition is specified in the transition, the transition occurs only when the condition is evaluated to true. Transitions that do not specify a conditions are always used.
- You can create multiple incoming transitions. This is called a **join** transition. As with split transitions, if you define conditions for each incoming transition, you can set a property on the activity that defines whether all conditions must be met (AND) or only one (XOR).

  Set the join type in an end block activity. The join type must be paired with previous split type. For an XOR join transition, the end block activity transitions when one of incoming transition is received. The XOR join transition is the default type.

  For an AND join transition, if the Join Count property in the end block is not defined, the end block activity will transition to the next activity when all of the incoming transitions are received. If Join Count is specified, the end block activity transition when the number of transitions received is equal to Join Count.
- A transition can use an exception expression defined by the engine to indicate that the transition occurs only when an exception is thrown by the incoming activity.

4

Reporting

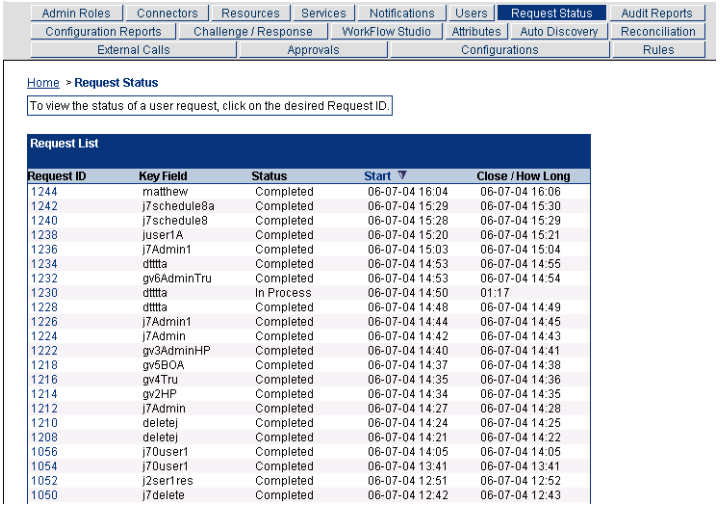
The Select Identity client enables you to view the status of workflow instances. Once a workflow instance starts, you can view its status, whether it is running or complete. You can view the status of an entire instance or the blocks in a workflow.

The format of the status report is defined in a report template. The report template describes what information is displayed, how it is organized, and the display format, and a report template can be shared by many workflow templates. By default, reports for all workflow instances use this default template.

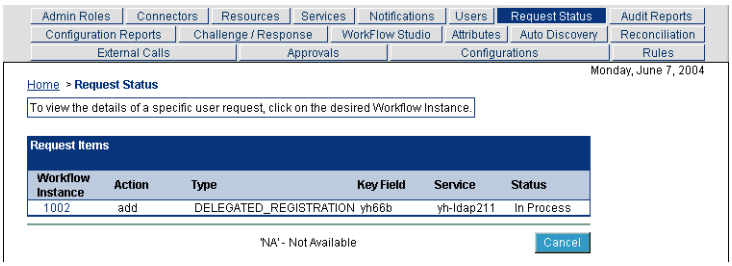
This chapter describes how to view a report for a workflow instance. It also describes the format of the report templates and how to create a custom report template for use by your workflow templates.

Viewing Workflow Status

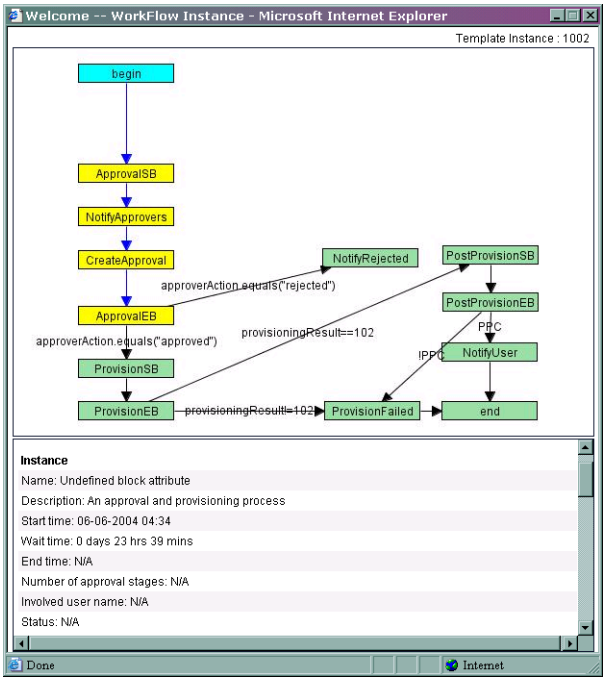
To view the status of a workflow, click **Request Status** on the Select Identity client and search for request instances. The following page displays:



To view the status of a workflow instance, click its ID in the table:



Click the workflow instance ID to display the report.



The colors used in the diagram indicate the status of each activity and transition:

- Green activities are complete; they have been executed and have transitioned.
- Blue transitions are complete and the workflow has transitioned past it.
- Yellow activities are waiting.
- White activities and black transitions have not been executed.

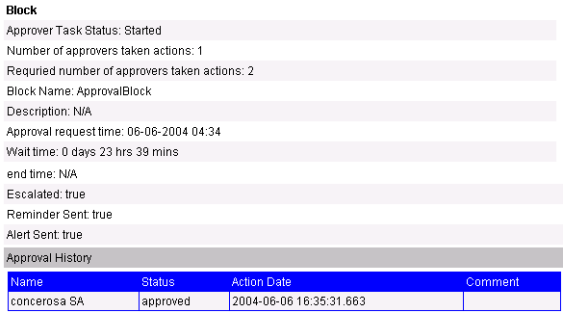
The tables below the diagram describe details about the entire workflow instance and its blocks. The content of these tables is determined by the default report template provided by Select Identity.

Here is an example of workflow instance status:

Instance

Name: Undefined block attribute
Description: An approval and provisioning process
Start time: 06-06-2004 04:34
Wait time: 0 days 23 hrs 39 mins
End time: N/A
Number of approval stages: N/A
Involved user name: N/A
Status: N/A

Here is an example of block status:



Block

Approver Task Status: Started
Number of approvers taken actions: 1
Requried number of approvers taken actions: 2
Block Name: ApprovalBlock
Description: N/A
Approval request time: 06-06-2004 04:34
Wait time: 0 days 23 hrs 39 mins
end time: N/A
Escalated: true
Reminder Sent: true
Alert Sent: true

Approval History

Name	Status	Action Date	Comment
concerosa SA	approved	2004-06-06 16:35:31.663	

Report Templates

A report template is associated with a workflow template using the reportType property specified on the Activity tab in Workflow Studio. Any workflow template whose report type matches the report template name uses the matching report template to render the status report. If the reportType is not defined in a workflow template, the default report template provided by Select Identity is used for that workflow template.

In addition to workflow-level reporting, the report template also describes how block status is reported. The workflow template uses the Block Type property to identify how a block is rendered. The block is rendered in a format given in the matching block section (identified by Type attribute in the <Block> element) of the associated report template.

In the default report template, three block types are defined that enable you to view block status:

- emailVerify
- approval
- provisioning
- postprovisioning
- ReconciliationPostProvision

If you assign one of these block types to the Block Type property in your workflow template, the report engine will render the information for the block according to the format defined in the default report template.

The following sections describe the format and contents of the report template file and provide steps for creating a new template based on the default.

Template Structure

A report template is organized hierarchically. At the highest level, there is one workflow instance and many blocks. Here is an overview of the XML file:

```
<Page>
  <Instance>
  ...
  </Instance>
  <Blocks>
    <Block>
    ...
    </Block>
  </Blocks>
</Page>
```

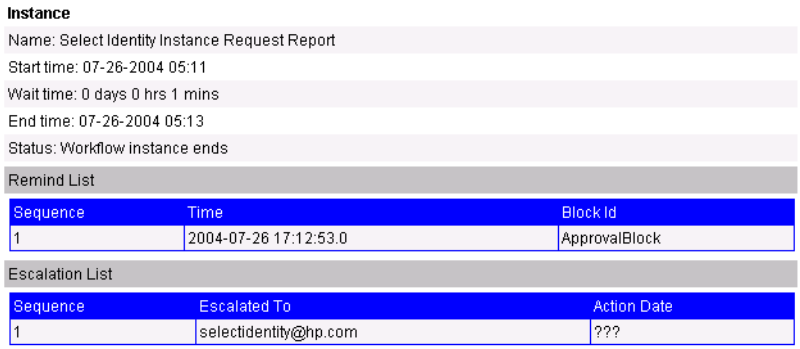
The <Instance> block defines properties for rendering workflow instance-level information. Each <Block> block defines how to render the named block in the workflow instance. The following sections describe the properties that define how the instances and blocks are rendered in the report.

Instance-level Reporting

The <Instance> block in the template defines how Select Identity renders workflow instance-level data. Here is an example of the <Instance> block from the default report template:

```
<Instance>

  <Row>
    <Text Name="Name" Value="Select Identity Instance Request Report"/>
  </Row>

  <Row>
    <Text Name="Start time" Value="StartTime" Scope="Eng"/>
    <Text Name="Wait time" Value="WaitTime" Scope="Eng"/>
    <Text Name="End time" Value="EndTime" Scope="Eng"/>
  </Row>

  <Row>
    <Text Name="Status" Value="Status" Scope="Eng"/>
  </Row>

  <Row>
    <Table Name="Remind List" Variable="remindList" VariableType="List"
     IgnoreNull="true">
      <Column ColLabel="Sequence" ColId="Attempts"/>
      <Column ColLabel="Time" ColId="RemindingTime"/>
      <Column ColLabel="Block Id" ColId="Block Id"/>
    </Table>
  </Row>

  <Row>
    <Table Name="Escalation List" Variable="escalationList"
     VariableType="List" IgnoreNull="true">
      <Column ColLabel="Sequence" ColId="Attempts"/>
      <Column ColLabel="Escalated To" ColId="EscalateTo"/>
      <Column ColLabel="Action Date" ColId="ActionDate"/>
      <Column ColLabel="Comment" ColId="ApproverComment"/>
    </Table>
  </Row>

  <Row>
    <Table Name="Exception List" Variable="exceptionInfoList"
     VariableType="List" IgnoreNull="true">
      <Column ColLabel="ID" ColId="Key"/>
      <Column ColLabel="Activity" ColId="Activity"/>
      <Column ColLabel="Exception" ColId="ExceptionClassName"/>
      <Column ColLabel="Message" ColId="Message"/>
      <Column ColLabel="Time" ColId="Time"/>
    </Table>
  </Row>

</Instance>
```

Here is an example of instance-level data displayed in a report according to these settings in the default report template:



Instance

Name: Select Identity Instance Request Report

Start time: 07-26-2004 05:11

Wait time: 0 days 0 hrs 1 mins

End time: 07-26-2004 05:13

Status: Workflow instance ends

Remind List

Sequence	Time	Block Id
1	2004-07-26 17:12:53.0	ApprovalBlock

Escalation List

Sequence	Escalated To	Action Date
1	selectidentity@hp.com	???

The <Instance> block in the report template defines five rows of text, as specified by the <Text> elements:

```
<Row>
  <Text Name="Name" Value="Select Identity Instance Request Report"/>
</Row>

<Row>
  <Text Name="Start time" Value="StartTime" Scope="Eng"/>
  <Text Name="Wait time" Value="WaitTime" Scope="Eng"/>
  <Text Name="End time" Value="EndTime" Scope="Eng"/>
</Row>

<Row>
  <Text Name="Status" Value="Status" Scope="Eng"/>
</Row>
```

The first row is named **Name** and its value is **Select Identity Instance Request Report**. You can see this text in the first row of the report and in the first <Text> element in the XML.

The report template also defines three tables — Remind List, Escalation List, and Exception List. For each table defined in the XML, columns are defined. For instance, the Remind List table has three columns — Sequence, Time, and Block Id:

```
<Table Name="Remind List" Variable="remindList" VariableType="List"
 IgnoreNull="true">
  <Column ColLabel="Sequence" ColId="Attempts"/>
  <Column ColLabel="Time" ColId="RemindingTime"/>
  <Column ColLabel="Block Id" ColId="Block Id"/>
</Table>
```

You can see these columns in the Remind List table in the report shown above.

Note that only two tables are displayed in this report. No exceptions occurred in the workflow instance so no Exception List table is displayed.

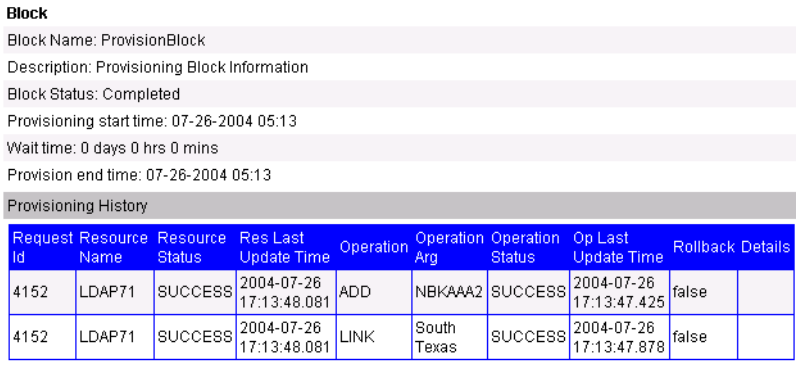
See <Text> Element on page 79 and <Table> Block on page 81 for details about configured the data to be displayed in reports.

Block-level Reporting

A report template can include multiple <Block> elements in the <Blocks> block to indicate how to render data for blocks in the workflow. The Type attribute defined in the <Block> element identifies the block type. The Block Type property set in the workflow must match this name in order for the report engine to render information according to the definition. Here is an example of a <Block> element from the default report template. This XML defines how to render information from the provisioning block:

```
<Block Type="provisioning">
  <Row>
    <Text Name="Block Name" Value="endBlockId" Scope="EA"/>
    <Text Name="Description" Value="Provisioning Block Information" />
  </Row>
  <Row>
    <Text Name="Block Status" Value="Status" Scope="Eng"/>
  </Row>
  <Row>
    <Text Name="Provisioning start time" Value="StartTime" Scope="Eng"/>
    <Text Name="Wait time" Value="WaitTime" Scope="Eng" />
    <Text Name="Provision end time" Value="EndTime" Scope="Eng" />
  </Row>
  <Row>
    <Table Name="Provisioning History" Variable="provisioningStatReport"
     VariableType="List">
      <Column ColLabel="Request Id" ColId="RequestId"/>
      <Column ColLabel="Resource Name" ColId="ResourceName"/>
      <Column ColLabel="Resource Status" ColId="ResourceStatus"/>
      <Column ColLabel="Res Last Update Time" ColId="ResLastUpdateTime"/>
      <Column ColLabel="Operation" ColId="OperationId"/>
      <Column ColLabel="Operation Arg" ColId="OperationArg"/>
      <Column ColLabel="Operation Status" ColId="OperationStatus"/>
      <Column ColLabel="Op Last Update Time" ColId="OpLastUpdateTime"/>
      <Column ColLabel="Rollback" ColId="ResRbFlag"/>
      <Column ColLabel="Details" ColId="Details"/>
    </Table>
  </Row>
</Block>
```

Here is a snapshot of how the Select Identity client render the provisioning block data in a workflow instance report:



Block

Block Name: ProvisionBlock

Description: Provisioning Block Information

Block Status: Completed

Provisioning start time: 07-26-2004 05:13

Wait time: 0 days 0 hrs 0 mins

Provision end time: 07-26-2004 05:13

Provisioning History

Request Id	Resource Name	Resource Status	Res Last Update Time	Operation	Operation Arg	Operation Status	Op Last Update Time	Rollback	Details
4152	LDAP71	SUCCESS	2004-07-26 17:13:48.081	ADD	NBKAAA2	SUCCESS	2004-07-26 17:13:47.425	false	
4152	LDAP71	SUCCESS	2004-07-26 17:13:48.081	LINK	South Texas	SUCCESS	2004-07-26 17:13:47.878	false	

As in the <Instance> block, the <Text> elements define the rows that are displayed in the report and the <Table> elements define the tables. See <Text> Element on page 79 and <Table> Block on page 81 for a description of the elements and attributes you can set for block-level reporting.

<Text> Element

The <Text> element defines data to be displayed in a text row in the report. This element must be created within a <Row> block, and you can specify <Text> elements in the <Instance> and <Block> blocks. Here are several examples from the default report template:

```
<Instance>
  <Row>
    <Text Name="Name" Value="Select Identity Instance Request Report"/>
  </Row>
  <Row>
    <Text Name="Start time" Value="StartTime" Scope="Eng"/>
    <Text Name="Wait time" Value="WaitTime" Scope="Eng"/>
    <Text Name="End time" Value="EndTime" Scope="Eng"/>
  </Row>
  ...
</Instance>
```

```
<Blocks>
  <Block Type="emailVerify">
    <Row>
      <Text Name="Email verifcation Status" Value="Status" Scope="Eng"/>
    </Row>
    <Row>
      <Text Name="Block Name" Value="endBlockId" Scope="EA"/>
```

```
      <Text Name="Description" Value="Email Verification Block Information"
       Scope="Var" />
    </Row>
...
```

To understand how the report engine renders this element, see the explanation in Instance-level Reporting on page 76.

Here is a description of the attributes required by the <Text> element:

- **Name** — The text row label. For example, if you specify **Name="Name"**, a row named Name is rendered for instance-level data in the report .
- **Value** — The value assigned to the row. The value assigned to this attribute depends on the value assigned to the Scope attribute, as described below.
- **Scope** — The type of data displayed in this row. The following values are supported for this attribute:
  - **Text** (or if the Scope attribute is omitted from the <Text> element) — Displays static text. For example, if you specify the following element:

    <Text Name="Name" Value="Select Identity Instance Request Report" />

    This text is displayed in the Name row of the report, as shown in the default report example on page 77.
  - **Var** — Displays the value of the specified variable. Variables are created in the workflow instance by the Set Variable action or when a variable name is assigned to the return value of an application invocation.

    For example, if you created the appList variable in a block to contain the list of approvers returned by the GetApproversBySpecifiedRole application invocation, you can specify the following in a <Block> block in the report template:

    <Text Name="Approvers" Value="appList" Scope="Var" />

    If the <Text ... Scope="Var"> element is in the <Instance> block, the all workflow variables are reported. If the <Text ... Scope="Var"> element is in a <Block> block, the engine reports on variables in the current block first. If the variable does not exist in the current block, it tries to report on the same variable at the workflow-level.

- **EA** — Displays the value of the specified property. Instance-level properties are set on the Workflow Information tab, which is available when you click in the background of a template. Block-level properties must be set in the end activity of the block. For a list of block-level properties, see page 43 that you can use for this attribute.

  For example, if you wish to print the name of the block in the report, specify the following element:

  ```
  <Text Name=”Block Name” Value=”endBlockId” Scope=”EA” />
  ```

- **Eng** — Displays engine-calculated data; you must specify the name of the data. You can specify the following values for this attribute:
  - StartTime — The start time of the instance or block
  - EndTime — The end time of the instance or block
  - WaitTime — The total wait time of the instance or block (how long all wait activities took in the instance or block)
  - Status — The status of the instance or block
  - joinedCount — A block-level variable that specifies the number of provisioning actions that took place before the block exited

<Table> Block

The <Table> block displays a list of data objects in a table in the report. . Each table displayed in the report template is mapped from a list. An object element in the list represents a row in the table and an attribute (field) in the object represents a column.

Within each <Table> block, <Column> elements define the content of the table. You can configure the report engine to render collection (list) variables created by the workflow engine or those defined by the Add Item to List action (see page 67).

Here are several examples from the default report template:

```
<Instance>
  ...
  <Row>
    <Table Name="Exception List" Variable="exceptionInfoList"
     VariableType="List" IgnoreNull="true">
      <Column ColLabel="ID" ColId="Key"/>
      <Column ColLabel="Activity" ColId="Activity"/>
```

```
      <Column ColLabel="Exception" ColId="ExceptionClassName"/>
      <Column ColLabel="Message" ColId="Message"/>
      <Column ColLabel="Time" ColId="Time"/>
    </Table>
  </Row>
</Instance>
<Blocks>
  ...
    <Row>
      <Table Name="PostProvisioning Activities"
       Variable="ReconPostProvisionStatReport" VariableType="List">
        <Column ColLabel="Request Id" ColId="RequestId"/>
        <Column ColLabel="Resource Name" ColId="ResourceName"/>
        <Column ColLabel="Service Name" ColId="ServiceName"/>
        <Column ColLabel="User Name" ColId="ConceroUserId"/>
        <Column ColLabel="Operation" ColId="RequestType"/>
        <Column ColLabel="Operation Status" ColId="Status"/>
        <Column ColLabel="Message" ColId="Msg"/>
      </Table>
    </Row>
  </Block>
</Blocks>
```

Each field in the object can be displayed as a column in the table.

Attributes of the <Table> Element

The following describes the attributes of the <Table> element. Refer to the XML listed above for examples.

- **Name** — The title of the table.
- **Variable** — The variable whose values you wish to display in a table. Specify the name of the variable created by an external application or the Add Item to List action, or specify the name of a workflow engine-defined variable. If you wish to render an engine-defined variable, the following values are supported:
  - **remindList** — Lists information about notifications (reminders) sent during the workflow instance or block activities
  - **escalationList** — Lists information about escalations that occurred during the workflow instance or block activities
  - **exceptionInfoList** — Lists information about exceptions that occurred during the workflow instance or block activities
  - **alertList** —Lists information about alerts sent during the workflow instance or block activities

  - **pushList** — Lists approvers in the block; specify this value for tables in <Block> blocks only
  - **provisioningStatReport** — Lists the status of each provisioning request; specify this value for tables in <Block> blocks only
  - **ReconPostProvisionStatReport** — Lists the status of each reconciliation request; specify this value for tables in <Block> blocks only
- **VariableType** — The type of variable. You must specify **List** as the value of this attribute; this is the only type supported.
- **IgnoreNull** — Whether to render the table if no values are available. Set this attribute to **true** if you do not want to render an empty table.

Attributes of the <Column> Element

For each <Table> element, you must define one or more <Column> elements to configure the data that will be displayed in the table. The attributes of the <Column> element are as follows:

- **ColLabel** — The column name (the label of the column displayed in the report)
- **ColId** — The field name in the object. If you are rendering a list workflow variable set by an external application, the ColId attribute corresponds to an attribute name of a data object element in the list. If you are rendering an engine-defined variable, the following fields are supported. Refer to the default report template for examples:
  - For the remindList variable:
    - **Attempts**
    - **RemindingTime**
    - **Block Id**
  - For the escalationList variable:
    - **Attempts**
    - **EscalateTo**
    - **ActionDate**
    - **ApproverComment**

- For the exceptionInfoList variable:
  - **Key**
  - **Activity**
  - **ExceptionClassName**
  - **Message**
  - **Time**
- For the alertList variable:
  - **Attempts**
  - **EscalateTo**
  - **ActionDate**
  - **ApproverComment**
- For the pushList variable:
  - **ApproverName**
  - **Status**
  - **ActionDate**
  - **ApproverComment**
- For the provisioningStatReport variable:
  - **RequestId**
  - **ResourceName**
  - **ResourceStatus**
  - **ResLastUpdateTime**
  - **OperationId**
  - **OperationArg**
  - **OperationStatus**
  - **OpLastUpdateTime**
  - **ResRbFlag**
  - **Details**

- For the ReconPostProvisionStatReport variable:
  - **RequestId**
  - **ResourceName**
  - **ServiceName**
  - **ConceroUserId**
  - **RequestType**
  - **Status**
  - **Msg**

For example, the following XML specifies to render data from the remindList variable:

```
<Row>
  <Table Name="Remind List" Variable="remindList"
   VariableType="List" IgnoreNull="true">
    <Column ColLabel="Sequence" ColId="Attempts"/>
    <Column ColLabel="Time" ColId="RemindingTime"/>
    <Column ColLabel="Block Id" ColId="Block Id"/>
  </Table>
</Row>
```

This XML is rendered as follows in the report:

Remind List

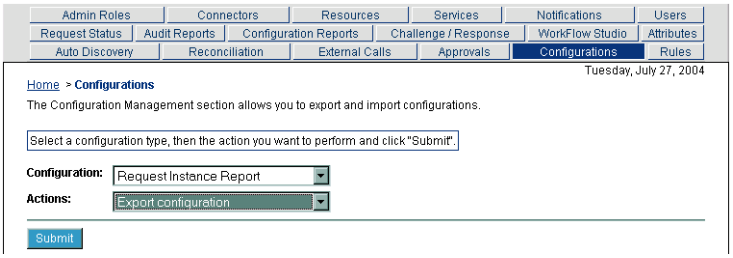
Sequence	Time	Block Id
1	2004-07-26 17:12:53.0	ApprovalBlock

Creating a Custom Report Template

A default report template is provided that describes how the report engine should render workflow data. You can export and modify the default template file from the Configurations home page on the Select Identity client. Or, you can create another report template. Complete the following steps to edit the report template in order to create your own:

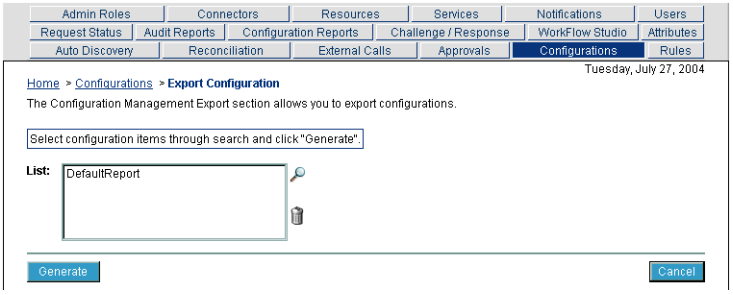
1. Export the default report template by completing these steps:
   a. On the Select Identity client, click **Configurations**.

b Select **Request Instance Report** from the Configuration drop-down list, select **Export configuration** from the Actions drop-down list, and click **Submit**.

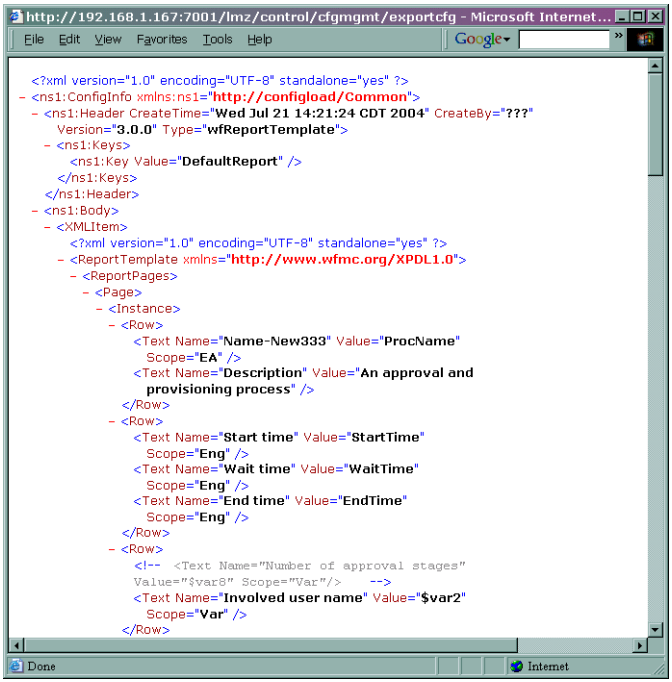


c Search for the file by clicking , submit the default search criteria, and select the **DefaultReport** file.

d Click **Add** then close the search window. The file is now listed on the Export Configuration page:



e Click **Generate** on the Export Configuration page. The file is displayed in a browser window.



f Save the file locally. Note that you may need to replace **<** with **<** and **>** with **>** in the file.

2 Edit the file as follows:

a Modify the elements in <Instance> block to define how workflow instance-level data will be rendered in the report. See Instance-level Reporting on page 76 for details about the elements and attributes in this block.

b If you are creating a new template file, you must change the Value attribute in the <ns1:Key> element. For example, if you wish to rename this template ReconciliationReport, change the following:

```
<ns1:Key Value="DefaultReport"></ns1:Key>
```

to

```
<ns1:Key Value="ReconciliationReport"></ns1:Key>
```

To assign this report template to a workflow template, be sure to create the reportType property as a global attribute for the template (on the template's Workflow Information tab, which you can display by clicking in the background of the template). Assign the name of the report template (such as ReconciliationReport in this example) as the value.

c. Modify the existing <Block> block, or add additional <Block> blocks, to define parameters for reporting on a specific blocks in your workflow template.

   To render a specific block in the workflow template according to a <Block> block in the report template, create the Block Type property in the end activity of the block. Assign the name of the <Block> block (as defined by the the Type attribute of the <Block> element) as the value of the property. See Block-level Reporting on page 78 for details.

d. Save the file.

3. Import the XML file into Select Identity:

   a. On the Select Identity client, click **Configurations**.

   b. Select **Request Instance Report** from the Configuration drop-down list, select **Import Configuration** from the Actions drop-down list, and click **Submit**.

   c. Browse for the file on your system then click **Submit**.

The XML file is loaded into the Select Identity database when you import it. The newly-defined report template is available in the Configurations drop-down list.

A

Frequently Asked Questions (FAQ)

General

What is the process for designing and creating a workflow template?

Follow these guidelines to design and create a workflow template:

1. Determine what tasks to create, such as for approval and provisioning.
2. Design a block for each task.
3. Determine what applications or actions must be invoked.
4. Decide how these applications are called (such as their order and the business rules that govern them) to layout activities and place actions.
5. List the input parameters required by the applications.
6. Determine what workflow variables are needed and how to create and pass them between activities.

What is a wait instance?

When a running workflow instance hits a wait activity, it is passivated until it is reactivated by an external source. The passivated workflow instance is called wait instance.

How is a workflow instance started (created)?

A workflow instance can be started using the workflow API (see the *HP OpenView Select Identity External Call Developer Guide* for details) or sub-workflow action invocation.

Are macros defined for use in a workflow expression?

A workflow macro is a function defined by the workflow engine to be used as part of an expression. For example, in a transition condition, you can use the equal() macro to compare two objects, as in this example: equal(approved,"approved"). The workflow engine will evaluate it and make a decision based on the returned result. See Using Variables on page 48 for the list of macros provided for workflow templates.

Where can I use workflow engine-defined variables?

The workflow engine internally defines a set of variables for advanced usage. These are low-level variables and only advanced users may access these variables. For example, when you register an application, you may need to pass the variables to application. See the *HP OpenView Select Identity External Call Developer Guide* for a list.

What is an asynchronous conversation in a workflow?

An asynchronous conversation consists of an asynchronous application invocation and a callback operations. It simulates what a synchronous invocation does – call and get a return value. In an asynchronous conversation, the return value is obtained when the external system reactivates the workflow by passing in workflow variables. The typical example can be found in a provisioning stage. Because provisioning can be a lengthy process, the workflow invokes it asynchronously and the provisioning later passes its status back to the workflow after it completes. This reactivates the workflow.

How can I implement an asynchronous conversation in a workflow template?

The conversation cycle starts with an asynchronous invocation to an external application. The conversation context is passed to the invoked application an input parameter, either using _instanceActivityId or using a combination of $_instId and _activityId. The context uniquely identifies where the reactivation point is in the workflow. The workflow is then passivated (in a wait activity). The activity can then be invoked by the external application with workflow variables set as input parameters. The workflow uses the variables as return values from the application.

Activities

What is the purpose of a wait activity?

A wait activity is like normal activity except that it does not automatically transition to the next activity. After all of the actions are executed, this activity suspends the running workflow instance temporarily until it is resumed later by an external application, such as by the Workflow API.

Typically, the _instActivityId variable that is defined by the workflow engine is passed to the external application, which uses this identifier to reactivate the workflow later. When the instance is reactivated, it resumes at a point where wait activity completes.

Does the catchException activity catch any exceptions thrown by any application?

Yes. It is a catchException activity transitions to when any exception is thrown from anywhere in the workflow.

When should an exception expression be used for transition's condition?

The catchException activity is used to globally catch an exception, while a transition with an exception condition enables the workflow to catch an exception at the activity level. See one of the default templates for an example In the default templates, the catchException can catch an exception thrown by an activities. It then transitoions to the Notify Error activity. If this activity throws another exception, an infinite loop will occur. To avoid this, use a transition with an exception condition as shown in the template.

Blocks

How can I create a block in a workflow template?

A block consists of a start block activity, an end block activity, and other activities between these two activities. Start and end block activities are identified by the same ID, which is assigned to the Start block Id and End Block Id properties.

How do I define block properties?

Block properties are defined in the end block activity.

How can I access block variables?

Block variables are accessed using Workflow API calls. A block variable is identified by name and block ID. A block variable can also be referenced in a report template to render the block information in the resulting report.

What does the Join Count property do in an end block activity?

The workflow engine uses the Join Count property defined in an end block activity to determine whether the end activity can exit. Internally, it stores a variable that counts the number of times an end block activity is executed. If the Join Count value matches the joined count, the end activity transitions out of the block. If Join Count is not specified, the incoming transition count for the end block is implicitly assumed to be the join count.

How do I make activities run concurrently?

Activities that are split by a start block activity with the AND join type run concurrently.

How do I to synchronize concurrently running activities?

Activities that join at an end block activity with AND join type transitions are synchronized. The end block activity does not exit until all transitions are received.

Is an end block activity a waiting activity?

An end block activity cannot be a wait activity. However, an end block activity can suspend the instance if the total number of anticipated transitions are not received.

Can a block time out?

A block can time out if it does not complete the block activities within the specified timeout property value. This timeout value is defined in the end block activity by the Timeout After property. You can also set the Timeout At property to time out at a specific time.

Actions

When should I use a subworkflow?

Subworkflows enable you to reuse of common workflow templates. For example, a multi-stage approval process can be defined as a single approval template and a parent workflow template can invoke the single approval subworkflow multiple times with different input parameter settings.

How does a parent workflow exchange variables with a subworkflow?

All persistent variables in the parent workflow are automatically passed to the subworkflow and all persistent variables from the subworkflow are passed back to the parent workflow when subworkflow returns. Engine-defined variables are not shared between workflow instances.

How does the workflow behave when a subworkflow contains wait activities?

When a subworkflow is invoked synchronously, the parent workflow instance will not continue until the subworkflow returns. For a subworkflow with wait activities, the Call Subworkflow action should be invoked by creating it in a wait activity or by selecting the subworkflow wait check box for the activity. The parent workflow resumes when the subworkflow completes.

Does a subworkflow run in the same instance as the parent workflow?

No, it runs in a separate instance.

Does the Throw Exception action terminate a workflow instance?

If a catchException activity is implemented, the instance will start at this activity right after the exception is thrown. Otherwise, the workflow instance propagates the exception back to the workflow client (where the workflow was invoked) and terminates itself.

Should I implement a catchException activity when the Throw Exception action is used?

catchException is optional. If it is not used, the thrown exception will be propagated to the workflow invoker (such as the Select Identity server).

How is the Error Recover action used?

This action is typically used with a catchException activity. The catchException activity is usually followed by a wait activity that forwards an alert message containing the exception context (getCurrentException() macro) to administrators who can fix the error. Once the error is fixed, the administrator can callback to the workflow using the exception context to resume the workflow which, in turn, uses the exception context to transition to activity that invokes the Recover From Last Error action. This action resumes the activity that threw the exception.

How do I know when to pass variable names or expressions to an action?

You can determine if a parameter is defined as text, a variable name, or an expression by viewing the Actions tab. As a rule, a parameter is always defined as an expression for application input parameters and as a variable for application return value.

Do engine-defined actions throw exceptions?

No, engine-defined actions do not throw exceptions. They log errors if an error occurs.

How can I log an exception?

The workflow engine stores a list of exceptions in a workflow variable called $_exceptionList. The report template uses this variable to display a list of exceptions. Therefore, to log an exception in a workflow template, simply log this variable.

Is the value returned by an application automatically assigned to a workflow variable?

Yes. The Return Variable Name field for each application invocation specifies the variable name that will contain the return value.

Does an application return a value when it is invoked asynchronously?

No, an asynchronous invocation returns to the caller before the invoked application returns a value.

How does the External Call application differ from other applications?

The External Call application requires that the application be implemented following certain conventions. See the *HP OpenView Select Identity External Call Developer Guide* for more information.

Reports

How do I edit a workflow report template?

You can download the default report template from the Configurations home page. After editing the template, upload the file. The uploaded report template takes effect immediately; you do not need to restart the server.

How can I render a variable or property in a report?

To render variable and property information in a report, configure the report template using the scope attribute of the <Text> element. See Creating a Custom Report Template on page 85 for more information.

What statistical information is defined by the report engine?

The following data can be rendered in a report:

- Start time, wait time, and end time
- Execution status
- IsAlertSent
- IsReminderSent
- IsEscalationSent
- Escalation List
- Reminder List
- Alert List
- Error List

B

Scenario: Enforcing Entitlement Rules

This appendix provides an example of how you can use Select Identity to enforce business rules. Using an external call as a step in a workflow, you can enforce entitlement rules. Consider the following scenario:

> A change to a user's department comes in through Reconciliation. As a result, Reconciliation runs a workflow, which checks the user's existing department and compares it to the department sent to Reconciliation. Based on the new department value, the old cost center and entitlement are removed from the user and the new values are added. (The resource is Active Directory.)

The following procedure illustrates how to register the external call that will evaluate whether the user's department has changed and assign new values, if necessary. The following is assumed about this scenario:

- LDAP is the resource in which users are provisioned for this example, thus it is assumed that the LDAP connector, resource, and attributes exist.
- The Service exists and the ReconciliationDefaultProcess workflow template is associated with the Service views.

Complete the following steps to create the external call and associate it with a workflow template:

1 Code the external call that will enforce the entitlement rules. As described in the *HP OpenView Select Identity External Call Developer Guide*, you must create a Java object that performs the task as part of workflow. This enables you to integrate approval processes with external processes and systems. You must use the External Call API and Workflow API, which define Java-based interfaces for creating external callouts.

In the script, the following codes are used:

Department	Entitlement	CostCenter
Sales	SA-505	101
Finance	FIN-505	205
HR	HR-101	308
Corporate	CORP-3	409

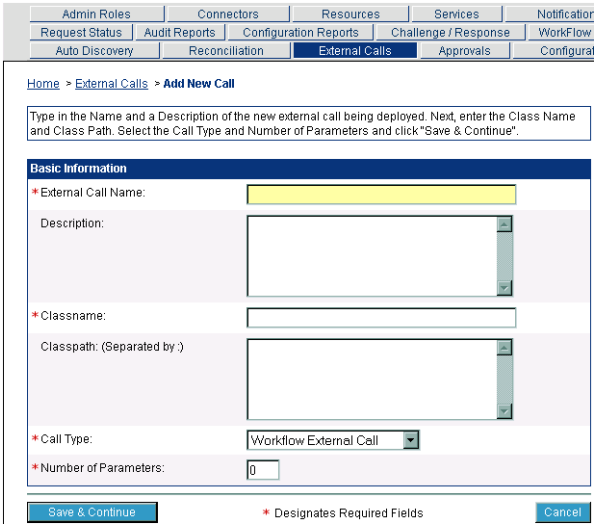
The script used for this scenario is provided as an example in the developer guide, in the Examples chapter. Here is an excerpt of that script:

```
//set the resource name here
private static final String resourceName = "LDAP_70";
private static final String USERNAMEATTRNAME = "UserName";
private static final String DEPARTMENTATTRNAME = "Department";
private static final String COSTCENTERATTRNAME = "CostCenter";
private static final String ENTITLEMENTPOSTFIX = "_ENTITLEMENTS";
private static final String findOldDeptQuery = "select A.stringValue
from TAAttribute A , "
            + "TAUser B where A.identObjId = B.userId AND A.name = "
            + "?  and B.conceroUserId = ?";

/*
 * If the department has changed, assign the new cost center and
 * entitlements to the user. Use the following table to assign values.
 * Department Entitlements CostCenter
 * ---------- ------------ -----------
 * Sales      SA-505       101
 * Finance    FIN-505      205
 * HR         HR-101       308
 * Corporate  CORP-3       409
 */

private static final String[] deptNames =
    {
    "Sales", "Finance", "HR", "Corporate"};
```

```
private static final String[] costCenterNames =
    {
    "101", "102", "103", "100"};

private static final String[] entNames =
    {
    "SA-103", "FIN-101", "HR-102", "Corp-103"};
```

2 Deploy the external call with Select Identity:

a From the External Calls home page in the Select Identity client, click **Add New Call**. The Basic Information page displays.



b Enter **`Entitlement Rules`** in the External Call Name field.

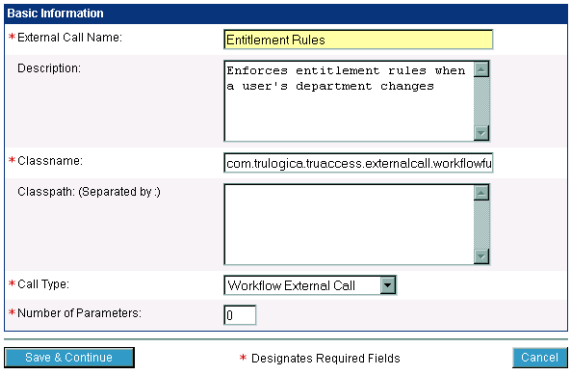
c Enter **`Enforces entitlement rules when a user's department changes`** in the Description text box.

d Enter **`com.trulogica.truaccess.externalcall.workflowfunction.WfEntitlementChange`** in the Classname field. This is the name of the class that implements the Java interface.

e Select **Workflow External Call** from the Call Type drop-down list.

f Enter **0** in the Number of Parameters field. The script does not require input parameters.

The Basic Information page now looks like this:

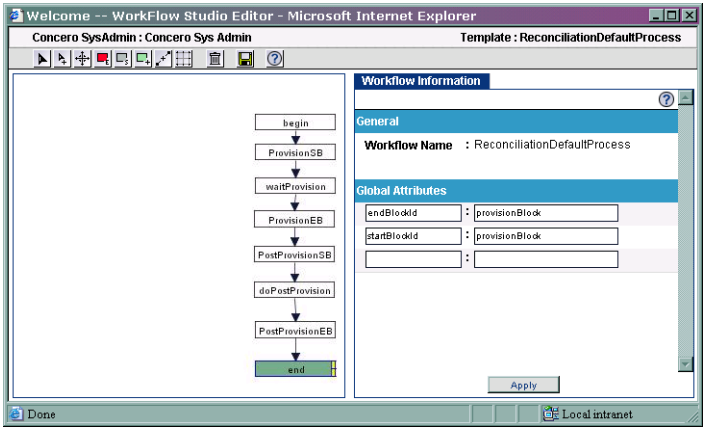


g Click **Save & Continue**.

For more information about deploying external calls, see the External Calls chapter in the *HP OpenView Select Identity Administrator Guide*.

3 Edit the default Reconciliation workflow template to call the Entitlement Rules external call.

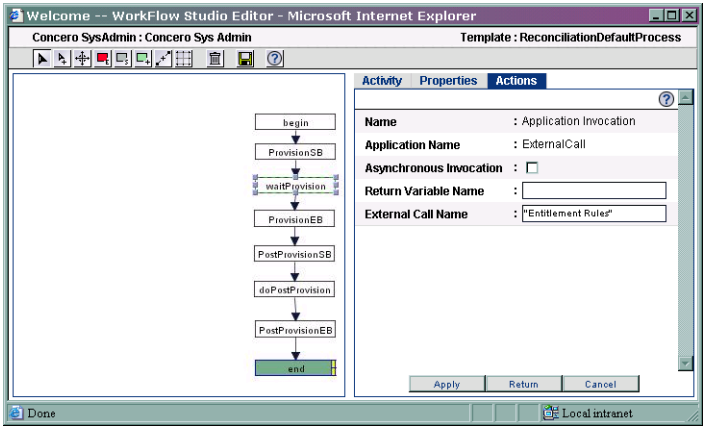
a On the Workflow Studio home page, select **ReconciliationDefaultProcess** from the Template drop-down list, select **Modify Workflow Template** from the Actions drop-down list, and click **Submit**. The Workflow Studio displays and loads the template.



b Modify the waitProvision activity to call the Entitlement Rules external call. Click on the toolbar and click the waitProvision activity. The property tabs are displayed on the right.

c Click the **Actions** tab then click the **Add** button.

d Select **Application Invocation** from the Name drop-down list, select **ExternalCall** from the Application Name drop-down list, enter **`Entitlement Rules`** in the External Call Name field, and click **Apply**.



e Save the template by clicking on the toolbar, and close Workflow Studio.

For more information about workflows, see the Workflow Studio chapter in the *HP OpenView Select Identity Administrator Guide*.

Now, Select Identity can enforce business rules. The external call is registered with Select Identity and the workflow will process the necessary entitlement changes based on changes to users.

C

Event Reference

The following tables lists the request events to which you can assign a workflow template when creating Business Relationships:

Delegated-registration Request Events

- Viewing a service membership
- Adding a user
- Modifying a user
- Deleting a service membership
- Adding a service to a user
- Enabling a service membership
- Disabling a service membership

Reconciliation Request Events

- Adding a service to a user
- Deleting a service membership
- Disabling all services

glossary

A

Access Control List (ACL)

An abstraction that organizes entitlements and controls authorization. An ACL is list of entitlements and users that is associated with a secured object, such as a file, an operation, or an application. In an ACL-based security system, protected objects carry their protection settings in the form of an ACL.

Access Management

The process of authentication and authorization.

Action

A task that can be performed within each Select Identity capability.

In Workflow Studio, an action invokes functions provided by the workflow engine or external applications within an activity. For example, you can log information to a file, set a property to be used later in the workflow, call an external process, provision a user in Select Identity, or store data in a database.

See also: Capability

Activity

A task that may occur when a workflow template is executed (in Workflow Studio). Activities are the core components of workflow templates; they do the work necessary to provision users. An activity can set a property to be used throughout the workflow, track approvals, start a subworkflow, send email, call an external application, and so on.

Admin Role

A template that defines the administrative actions that can be performed by a user. An Administrative Service is created to provide access to roles. Users are then given access to the Service. Users with administrative roles can also grant their set of roles to another administrator within their Service context.

Approval Process

The process of approving the association, modification, or revocation of entitlements for an identity. This process is automated of these through workflow templates.

Approver

A Select Identity administrator who has been given approval actions through an Admin Role.

Attribute

An individual field that helps define an identity profile. For each identity, an attribute has a corresponding value. For example, an attribute could be "department" with possible values of "IT," "sales," or "support."

Audit Report

A report that provides regular account interaction information within the Select Identity system.

Authentication

Verification of an identity's credentials.

Authoritative Source

A resource that has been designated as the "authority" for identity information. Select Identity accounts can be reconciled against accounts in an authoritative source.

Authorization

Real-time enforcement of an identity's entitlements. Authentication is a prerequisite for authorization.

Auto Discovery

The process of adding user accounts to the Select Identity system for a specified Service through the use of a data file.

B

Block

A special type of activity that serves two purposes: to define information to be used by a subset of activities (block-level properties) and to provide block-level reporting. For example, you might define a block that submits an approval request, waits for the response, and returns the status of the request to the workflow. In other words, think of a block as a process within a template.

Block Type

A property that is assigned to a block in a workflow template using the Block Type property in end block activity. The report template uses this property to identify how block information is rendered in the resulting report.

Business Relationship

A Select Identity abstraction that defines how a logical grouping of users will access a Select Identity Service. The Select Identity Service is a superset of all the identity management elements of a business service.

Business Service

A product or facility offered by, or a core process used by, a business in support of its day-to-day operations. Example business services could include an online banking service, the customer support process, and IT infrastructure services such as email, calendaring, and network access.

See also: Service

C

Capability

Actions that can be performed within the Select Identity client are grouped by capability, or link, in the interface.

See also: Action

Challenge and Response

A method of supplying alternate authentication credentials, typically used when a password is forgotten. Select Identity challenges the end user with a question and the user must provide a correct response. If the user answers the question correctly, Select Identity resets the password to a random value and sends email to the user. The challenge question can be configured by the administrator. The valid response is stored for each user with the user's profile and can be updated by an authenticated user through the Self Service pages.

Configurations

A capability that enables you to import and export Select Identity settings and configurations. This is useful when moving from a test to a production environment.

Configuration Report

A report that provides current system information for user, administrator, and Service management activities.

Connector

A J2EE connector that communicates with the system resources that contain your identity profile information.

Context

A Select Identity concept that defines a logical grouping of users that can access a Service.

Contextual Identity Management (CIM)

An organizational model that introduces new abstractions that simplify and provide scale to the business processes associated with identity management. These abstractions are modeled after elements that exist in businesses today and include Select Identity Services and Business Relationships.

Credential

A mechanism or device used to verify the authenticity of an identity. For example, a user ID and password, biometrics, and digital certificates are considered credentials.

D

Data File

An SPML file that enables you to define user accounts to be added to Select Identity through Auto Discovery or Reconciliation.

Delegated Administration

The ability to securely assign a subset of administrative roles to one or more users for administrative management and distribution of workload. Select Identity enables role delegation through the Self Service pages from one administrator to another user within the same Service context.

Delegated Registration

Registration performed by an administrator on behalf of an end user.

See also: Self Registration

E

End User

A role associated to every user in the Select Identity system that enables access to the Self Service pages.

Entitlement

An abstraction of the resource privileges granted to an identity. Entitlements are resource-specific and can be resource account IDs, resource role memberships, resource group memberships, and resource access rights and privileges. Entitlements are also considered privileges, permissions, or access rights.

Expression

A combination of workflow variables and constant values to be evaluated. An expression can be assigned to a new variable or passed to an application as an argument. If you are familiar with a programming language, an expression used in a workflow template is like C or Java expression. Example of expressions can be found in action input parameters, application return values, and transition conditions. See Using Variables on page 48 for details.

External Call

A programmatic call to a third-party application or system for the purpose of validating accounts or constraining attribute values.

F

Form

An electronic document used to capture information from end users. Forms are used by Select Identity in many business processes for information capture and system operation.

I

Identity

The set of authentication credentials, profile information, and entitlements for a single user or system entity. Identity is often used as a synonym for “user,” although an identity can represent a system and not necessarily a person.

Identity Management

The set of processes and technologies involved in creating, modifying, deleting, organizing, and auditing identities.

Instance

See: Workflow Instance

M

Management

The ongoing maintenance of an object or set of objects, including creating, modifying, deleting, organizing, auditing, and reporting.

N

Notifications

The capability that enables you to create and manage templates that define the messages that are sent when a system event occurs.

P

Password Reset

The ability to set a password to a system-generated value. Select Identity uses a challenge and response method to authenticate the user and then allow the user to reset or change a password.

Persistent Variable

A variable that is persisted after an instance is passivated. To extend the variable lifecycle to the entire instance, you must create the variable to be persistent. This enables the variable to be created before a wait activity, and it will be accessible after the workflow instance resumes. To make a variable persistent, precede the name with $. For example, the $retryCount variable is persistent while retryCount is not.

See also: Workflow Variable

Policy

A set of regulations set by an organization to assist in managing some aspect of its business. For example, policy may determine the type of internal and external information resources that employees can access.

Process

A repeatable procedure used to perform a set of tasks or achieve some objective. Whether manual or automated, all processes require input and generate output. A process can be as simple as a single task or as complicated a multi-step, conditional procedure.

See also: Approval Process

Profile

Descriptive attributes associated with an identity, such as name, address, title, company, or cost center.

Property

See:Workflow Property

Provisioning

The process of assigning authentication credentials to identities.

R

Reconciliation

The process by which Select Identity accounts are synchronized with a system resource. Accounts can be added to the Select Identity system through the use of an SPML data file.

Registration

The process of requesting access to one or more resources. Registration is generally performed by an end user seeking resource access, or by an administrator registering a user on a user's behalf.

See also: Delegated Registration, Self Registration

Request

An event within the Select Identity system for the addition, modification, or removal of a user account. Requests are monitored through the Request Status capability.

Resource

Any single application or information repository. Resources typically include applications, directories, and databases that store identity information.

Role

A simple abstraction that associates entitlements with identities. A role is an aggregation of entitlements and users, typically organized by job function.

See also: Admin Role

Rule

A programmatic control over system behavior. Rules in Select Identity are typically used for programmatic assignment of Services. Rules can also be used to detect changes in system resources.

S

Self Registration

Registration performed by an end user seeking access to one or more resources.

See also: Delegated Registration

Self Service

The ability to securely allow end users to manage aspects of a system on their own behalf. Select Identity provides the following self-service capabilities: registration, profile management, and password management (including password change, reset, and synchronization).

Service

A business-centric abstraction representing resources, entitlements, and other identity-related entities. Services represent the products and services that you offer to customers and partners.

Service Attribute

A set of attributes and values that are available for or required by a Service. Attributes are created and managed through the Attributes pages.

See also: Attribute

Service View

A restricted view of a Service that is valid for a group of users. Views enable you to define a subset of Service registration fields, change field names, reorder fields, and mask field values for specific users.

Single Sign-On (SSO)

A session/authentication process that permits a user to enter one set of credentials (name and password) in order to access multiple applications. A Web SSO is a specialized SSO system for web applications.

SPML Data File

See: Data File

T

Template

See: Workflow Template

Transition

The definition of a relationship between activities. You can define that one activity always follows another, or you can define a condition that must be met before the workflow transitions from an activity to one or more others. For example, you can define a transition that only allows the workflow to progress if at least two administrators approve a request. If the request is not approved, the workflow can transition to an activity that sends email notification to an administrator.

U

Users

The Select Identity capability that provides consistent account creation and management across Services.

V

Variable

See: Workflow Variable

Variable Expression

See: Expression

W

Workflow Engine

A system component that executes workflows and advances them through their flow steps.

Workflow Instance

An invocation of a workflow template. An instance starts when it is created and ends when it completes (when the last activity is executed). An instance's

status and other associated information can be viewed once an instance is created.

Workflow Process

The tasks, procedural steps, organizations or people involved, and required input and output information needed for each step in a business process. In identity management, the most common workflows are for provisioning and approval processes.

Workflow Property

A name-value pair, where the value is a text string. A property stores static data that cannot be changed at runtime. It can be accessed by the workflow API and report template. There are three levels of properties: global, block, and activity.

Workflow Studio

The Select Identity capability that enables you to create and manage workflow templates.

Workflow Template

A model of the provisioning process that enables Select Identity to automate the actions that approvers and systems management software must perform.

Workflow Variable

A name-value pair that can be created or changed at runtime in a workflow instance through actions, a workflow API call, or returned by an application invocation. It can be accessed by workflow API, workflow template, and report template. There are levels of variables: global, block, and activity.

See also: Persistent Variable

index

Symbols

A

B

C

D

E

F

G

J

L

N

O

P

R

S

T

V

W

X