Oracle Knowledge AnswerFlow Implementation and User’s Guide

A Guide to Implementing and Using Oracle Knowledge AnswerFlow

Oracle Knowledge AnswerFlow

Release 8.6

Document Number OKAF-IUSR86-00

August 2015

COPYRIGHT INFORMATION

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Contents

Preface About This Guide 1

In This Guide 1
Examples of Product Screens and Text 1
Operating System Variations in Examples and Procedures 2
References to Web Content 2

Overview of Oracle Knowledge AnswerFlow 3

About AnswerFlow Components 4
How AnswerFlow Fits in the Oracle Knowledge Architecture 5
Navigating To and Within the AnswerFlow Editor Web Application 6
Navigating to the AnswerFlow Editor Web Application 6
Navigating Within AnswerFlow Editor 7
Home Page 7
Promotions Page 8
Configurations Page 11
Data Migration Page 11
Using the AnswerFlow SampleUI Dashboard 12

Configuring and Administering AnswerFlow 14

Audience for This Section 14
Enabling AnswerFlow to Work With Information Manager 14
Working With the AnswerFlow Channel in Information Manager 14
Applying Categories to the AnswerFlow Channel 18
Connecting SampleUI to Information Manager 21
Enabling JGroups 21
Administering the AnswerFlow Editor Web Application 22
Managing AnswerFlow Properties and Access 22
Managing AnswerFlow Properties 22

Managing AnswerFlow Access Control 24
Setting Up Memcached 26
Migrating Data from Previous Versions of AnswerFlow 27
Starting and Stopping AnswerFlow 30
Enabling AnswerFlow to Work With Oracle Knowledge Analytics 30
Modifying JMS Properties Files to Send Events to Analytics 31
Troubleshooting AnswerFlow 31

Composing Process Flows 33
How to Navigate the Canvas 33
Leaving the Process Page 33
Actions in Edit Mode 34
Actions in View Mode 34
Editing Tools 35
Quick Search 36
Spelling Checker 36
About Process Objects, Page Elements, and Logical Nodes 36
Process Objects 36
Processes 36
Subprocesses 36
Service Calls 37
Context Variables 39
Configuring Pages and Page Elements 40
About Pages 40
About Page Elements 40
Configuring Display Form Page Elements 41
Configuring Form Control Parameters 42
Adding Display HTML Page Elements 43
Logical Nodes 43
Adding Branch Logical Nodes 43
Adding Service Call Logical Nodes 44
Adding Set Variable Values Logical Nodes 45
Adding Process Call Logical Nodes 45
Previewing a Process 45
Debugging a Process 46
Submitting a Process Object 46
Deleting a Process Object 46
Reverting a Process Object 46
Saving a PDF of the Canvas Window 46

An Example of Creating a Process Flow 47
Process Flow Best Practices 47
Overview of the Process Flow Example 47
Create Service Methods 48
Sample Java File 48

Create Three Service Methods 49
Create Context Variables 49
Create Three Service Calls 50
Create Service Call checkAccount 50
Create Service Call purchaseTime 50
Create Service Call signUpForService 51
Create the Subprocess Sign Up for Service 51
Create a Subprocess 52
Set a Variable 52
Create the Sign Up for Service Page 52
Add a Branch Node 54
Set a Variable 55
Add a Service Call 55
Validate Your Subprocess 56
Create the Top-Level Process HVAC Service 56
Create the Process 57
Start the Process with a Service Call 57
Confirm the Account is Current 58
Confirm the Product is Gold Eligible 59
Confirm Purchase is Gold Eligible 59
Create the Sign Up for Service Conditions and Dialog 61
Confirm Customer is Signed Up 62
Provide Service Knowledge By Level 62
Save, Preview, and Debug the Process 63

Deploying Processes 64

Compiling Processes 64
Promoting Processes 67
Re-Promoting Processes 67
Publishing Processes Within Information Manager 68
Unpublishing Processes Within Information Manager 70

Integrating Processes in End-User Interfaces 71

Using SampleUI to Deliver Process Flows to End-Users 71
Compiling and Deploying SampleUI 71
Embedding the SampleUI Dashboard in Web Applications 72
Requirements 72
Locate the Information Center Web application Files 72
Setting Up and Configuring SampleUI 72
Embedding Process Flows 73
Troubleshooting 73
Customizing the SampleUI 73
Requirement 73
About SampleUI Source Code 74
About Functional Mappings in SampleUI 76

Reviewing a Customization Example 77
Integrating Process Flows in Web Applications by Using JSON 78
Requirements 78
Executing Processes with /SampleUI/start 78
HTTP Method: GET or POST 78
Submitting a Form Within a Process by Using /SampleUI/step 79
HTTP Method: GET or POST 79
Information Manager Document 82
Information Manager Document Attribute 82
Information Manager Document Attribute Type 83
Information Manager Document Attribute Content 83
JSONP 86
Using Search for Process Integration 86
Searching for Process Flows by Using Managed Answers 86
Requirements 86
Using Search Results to Find Process Flows in the AF Channel 87
Using Search Results to Launch Process Flows .87
Launching Process Flows in the SampleUI 87
Launching Process Flows in Information Center 87

PREFACE

About This Guide

This user guide provides information to administer the AnswerFlow applications, create and maintain process objects and process flows, publish process flows, and integrate process flows into end-user web applications.

This preface contains the following information:

- "In This Guide"
- "Examples of Product Screens and Text"
- "Operating System Variations in Examples and Procedures"
- "References to Web Content"

In This Guide

The *Oracle Knowledge AnswerFlow Implementation and User's Guide* is divided into the following sections:

"Getting Started with AnswerFlow"	An overview of application capabilities and features.
"Configuring and Administering AnswerFlow"	Details for Oracle Knowledge application administrators who configure the AnswerFlow web application editor and runtime UI settings, manage the mapping of Information Manager Console roles to AnswerFlow functions (that is, manage access control), migrate data from Oracle Knowledge 8.5 to 8.6, and troubleshoot user or application problems.
"Creating and Editing Processes and Components"	Procedures for using the AnswerFlow web editor to create, edit, preview, validate, and submit process objects and process flows.
"Deploying Processes"	Procedures for promoting compiled process flows into Information Manager, where users can publish and unpublish process flows as needed within the AnswerFlow-specific IM channel.
"Integrating Processes in End-User Interfaces"	Procedures for developers to integrate published process flows into Information Center or custom service web applications by using the Sample UI reference application project. This chapter also has information on making process flows available by way of Search as Managed Answers or in search results of the crawled AnswerFlow-specific IM channel.

Examples of Product Screens and Text

The product screens, screen text, and file contents depicted in the documentation are examples. We attempt to convey the product's appearance and functionality as accurately as possible; however, the actual product contents and displays may differ from the published examples.

Operating System Variations in Examples and Procedures

We generally use Linux screen displays and naming conventions in our examples and procedures. We include other operating system-specific procedures or steps as noted in section headings, or within topics, as appropriate.

We present command syntax, program output, and screen displays:

- in Linux format first
- in other Unix-specific variants only when necessary for proper operation or to clarify functional differences
- in Windows format only when necessary for clarity

References to Web Content

For your convenience, this guide refers to Uniform Resource Locators (URLs) for resources published on the World Wide Web, when appropriate. We attempt to provide accurate information; however, these resources are controlled by their respective owners and are therefore subject to change at any time.

CHAPTER 1

Overview of Oracle Knowledge AnswerFlow

AnswerFlow is a decision-tree process modeling web application and process display runtime server. Knowledge management teams can use AnswerFlow to compose support inquiry process flows that deliver Oracle Knowledge documents based on process conditions. Process designers can troubleshoot problems by using business logic to aid agents through questions, options, diagnostics, and decisions.

This document uses the following terminology:

Term	Definition
AnswerFlow Editor	An editing tool in which an author can create, edit, and preview processes that describe their business logic. Processes are designed on a canvas page in which process elements can be dragged and dropped to the canvas and arranged or linked based on the decision paths that exist in the process.
Compile	An AnswerFlow Editor operation that is performed on a Top Level Process which performs validation as well as collecting all object data from its dependencies, such as subprocesses (and recursively its subprocesses) and service call definitions. This action generates a promotion record.
Conditional Branch	An AnswerFlow Step where a decision is made as to which step to execute next. The decision is based on the values of Local and Context Variables.
Context Variable	A variable which is available to all processes in AnswerFlow. Its value is set prior to the execution of a process and does not change during execution.
Debug Mode	A mode of executing a process in AnswerFlow Editor which allows the author to view both the user-facing activities and the programmatic information (for example, conditional logic, variables, and service calls).
Instance	A server container in which an application runs.
Local Variable	A variable that is visible only to the process.
Preview Mode	A mode of executing a process in AnswerFlow Editor which allows the author to view it as a process user would see it from a production end-user UI.
Promote	An AnswerFlow Editor operation that is performed on a Compiled Process. Promote pushes the compiled data to an Information Manager document, creating a new document if one does not exist.
Save	An AnswerFlow Editor operation to persist the current flow.
Save As	An AnswerFlow Editor operation to persist the current flow, but as a new process. This process can then be modified and saved exclusively from the original process.
Service Call	A step in which work is delegated to a system outside of business logic defined in the AnswerFlow process. This work is typically more complex in nature. Some examples are performing webservice calls or performing arithmetic.
Step	In AnswerFlow a Step is one unit of work. It can be a conditional branch, page display with one or more page elements, variable assignment, subprocess call, or service call.

Term	Definition
Submit	An AnswerFlow Editor operation which, when performed on an Object (Process, Service Call Object, or Context Variable), validates the Object and, if successful, saves a snapshot. **Note:** Any referenced Context Variable, Service Call Object, or Subprocess must be submitted for validation to succeed.
Subprocess	A process which is entered from and returns to another process. A Subprocess cannot be a Top Level Process.
Subprocess call	A step in which a work is delegated to another process.
Top Level Process	A logical collection of steps which drive business logic. When compiled, it contains all the data from all its subprocesses (and recursively its subprocesses). A Top Level Process cannot be a Subprocess.

About AnswerFlow Components

AnswerFlow consists of the following main components:

- **Editor**, to create and manage AnswerFlow processes, subprocesses, context variables, and service calls
- **Datastore**, to store the AnswerFlow processes and process components
- **Runtime User Interface (RuntimeUI**), to test processes in development and provide a basis for integrating AnswerFlow processes with your production application

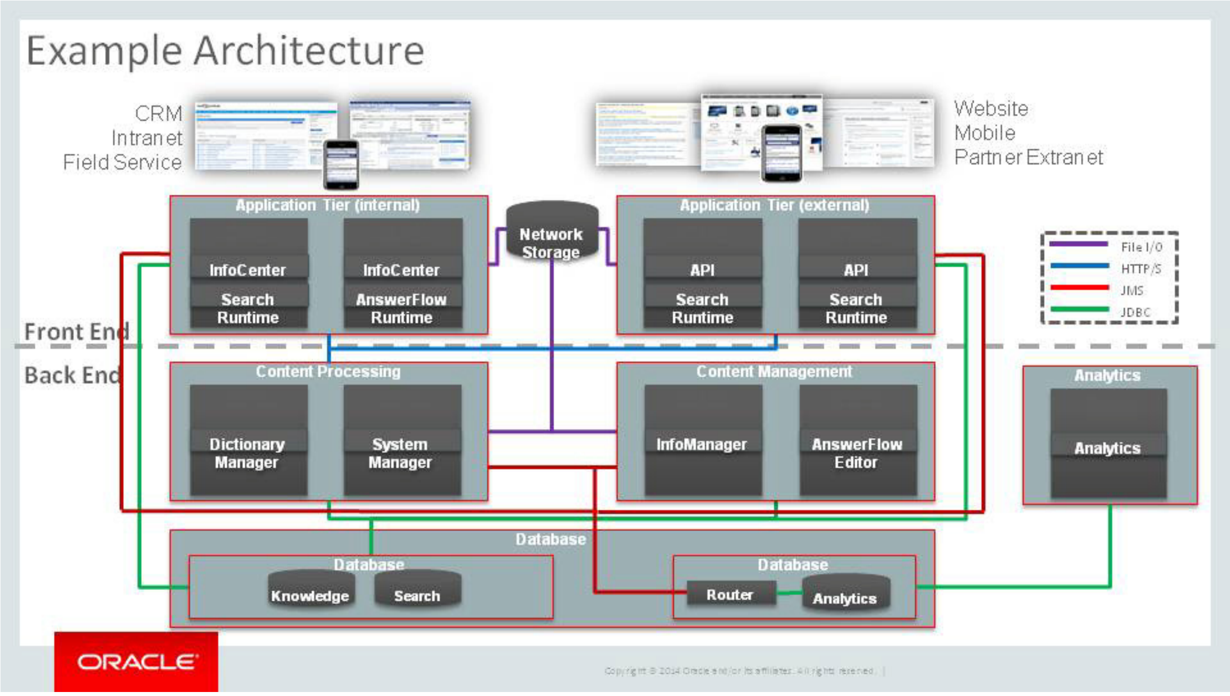
The following table describes AnswerFlow components and functions:

Component or Function	Description
AnswerFlow Editor Interface	AnswerFlow Editor is a web-based user interface that you install and configure as part of the standard installation. Use the editor to create and manage AnswerFlow processes and their components as described in “Creating and Editing Processes and Components”.
Global Search	The AnswerFlow Editor Home page has a Search field for finding objects by name. In the Search field, enter the object name or part of the object name and click the Search button. Supported wild cards include _=for any character) and "%" (for any sequence of characters). You can search only against the name attribute, that is, the process or object name or part of the name.
AnswerFlow Datastore of Objects	The AnswerFlow installation creates a dedicated datastore that is a set of tables within a database. The datastore initially contains sample AnswerFlow objects and will contain user-generated AnswerFlow objects, such as processes, context variables, and service calls. The AnswerFlow RuntimeUI accesses these objects and presents them in a sample web user interface. You can configure the AnswerFlow Editor and RuntimeUI instances in the developer, staging, and production environments to point to the datastore to access its contents.
AnswerFlow End-User RuntimeUI	The AnswerFlow installation includes the AnswerFlow RuntimeUI. RuntimeUI is the instance in which SampleUI (a sample user interface) is deployed ready-to-use. For instance, running `instances/RuntimeUI/setenv.sh` followed by `inquiraaf.sh start` starts the AnswerFlow RuntimeUI server (in `Tomcat/WebLogic/WebSphere`), which has SampleUI running on it (if `deploy.sh` has been run). You can customize the RuntimeUI to conform to the look and feel of your end-user facing application by using the processes described in “Integrating Processes in End-User Interfaces”.

Component or Function	Description
Objects and Categories	All objects are in a single category structure in Information Manager. The Information Manager categories determine how AnswerFlow arranges objects in the Object Library. The structure may include multiple levels in a hierarchy.
Category Assignments	For a category to be visible to a user, the category must be assigned to the AnswerFlow channel, and the category or a category above it hierarchically must be assigned to the user. The categories must meet the first, and one of the second and third requirements.
Object Versions	Every time you submit an object, AnswerFlow creates a new version of the object that you cannot delete. Previous versions of objects remain available for you to use. On the Home page, the user clicks the number in the Version column to list the versions. On the Promotions page, the promoted versions of each object are listed sequentially. When you submit a process, you create a set version of the process that you can reference and revert to later.
Process Search by Tag	After a process is promoted to Information Manager, the tags populated in the Information Manager document contain the tags of the promoted process revision. Within Information Manager, you can search for processes by tags. The AnswerFlow Home page includes a Tags column where users can edit tags.
Processes and Oracle Knowledge Information Manager	Promoted processes are stored as Information Manager documents in a single Information Manager channel. The AnswerFlow channel is the Information Manager channel where AnswerFlow stores data. After you set up this channel, you can promote processes to it.
Processes as Documents	Use the Promotions page to promote a process as an Information Manager document. When you promote a process, the next step is to publish the document so it is available to users. Before you publish a document containing a process, you can edit the document in AnswerFlow or preview it in Information Manager.
Process Previews	As you create a process, you can preview what the resulting execution will look like and modify the process accordingly.

How AnswerFlow Fits in the Oracle Knowledge Architecture

At a high level, in the Oracle Knowledge architecture, AnswerFlow has two components: the editor web application and the runtime UI server in the UI tier, as shown in the following figure.



For detailed technical architecture information, refer to *Installing and Configuring Oracle Knowledge* and the Oracle University course Oracle Knowledge Technical Architecture and Configuration.

Navigating To and Within the AnswerFlow Editor Web Application

This section describes how to navigate to the AnswerFlow Editor web application itself, then describes how to navigate within AnswerFlow after you sign in.

Navigating to the AnswerFlow Editor Web Application

Your application administrator provides the server name of your Oracle Knowledge instance. The URL guidance provided here assumes a standard, ready-to-use installation of AnswerFlow as part of an overall Oracle Knowledge instance. If this URL does not lead you to the AnswerFlow login screen, contact your application administrator.

To sign in:

1. Navigate to the AnswerFlow Editor web application by opening a browser window and typing: `http://<servername>:8232/AnswerFlowEditor/faces/login.jspx`
2. On the AnswerFlow Login page, enter your Information Manager username and password. You do not need to choose an Information Manager Repository.
3. Click **Submit** to open the AnswerFlow Home page.

Navigating Within AnswerFlow Editor

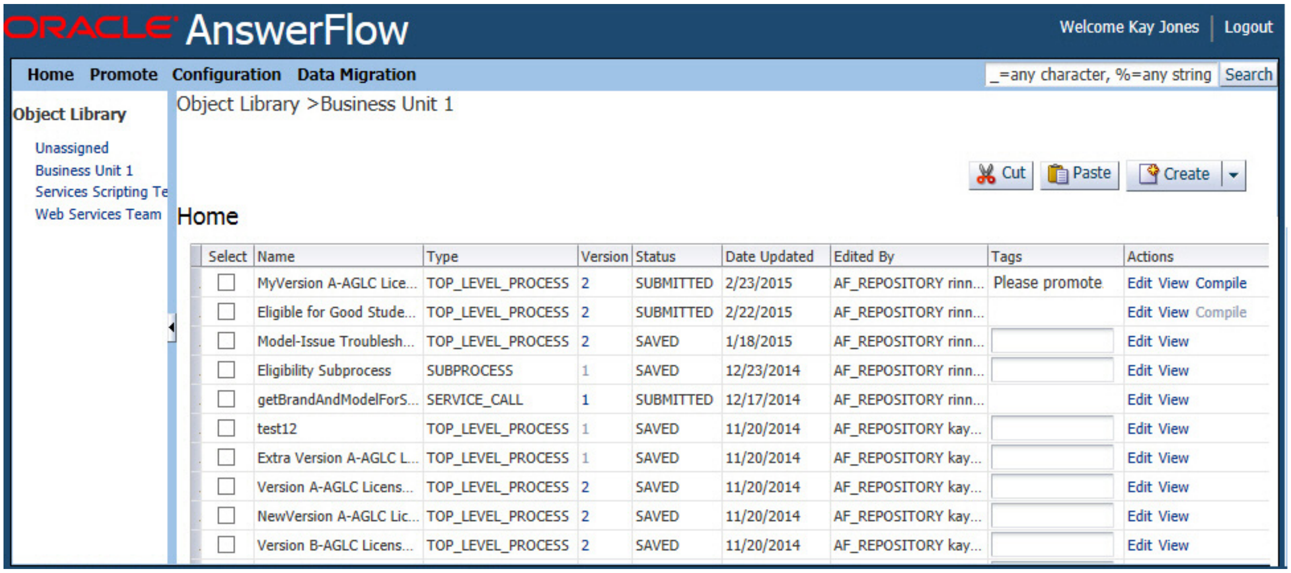
The Home page is your starting point in AnswerFlow Editor. From the Home page, you can navigate to Objects and other functionality.

Home Page

The global header has a darker top bar with the AnswerFlow logo on the left, and your user name and a Logout link is on the right.

The lighter bar of the global header has a search box on the right side of the screen. You can search by process name on both the Home and Promote pages, to find the process object of interest.

The global navigation links for the Editor web application are on the left of the global header, under the logo. The following figure shows the Home page. The left side of the screen has a column labeled Object Library. Categories listed here in the navigation tree correspond to the categories created for the AnswerFlow channel that is created in Information Manager. Your Information Manager role determines which categories are assigned to you.


ORACLE AnswerFlow
Welcome Kay Jones | Logout
Home Promote Configuration Data Migration
_=any character, %=any string
Search
Object Library
Unassigned
Business Unit 1
Services Scripting Te
Web Services Team
Object Library >Business Unit 1
Cut
Paste
Create
Home

Select	Name	Type	Version	Status	Date Updated	Edited By	Tags	Actions
□	MyVersion A-AGLC Lice...	TOP_LEVEL_PROCESS	2	SUBMITTED	2/23/2015	AF_REPOSITORY rinn...	Please promote	Edit View Compile
□	Eligible for Good Stude...	TOP_LEVEL_PROCESS	2	SUBMITTED	2/22/2015	AF_REPOSITORY rinn...		Edit View Compile
□	Model-Issue Troublesh...	TOP_LEVEL_PROCESS	2	SAVED	1/18/2015	AF_REPOSITORY rinn...		Edit View
□	Eligibility Subprocess	SUBPROCESS	1	SAVED	12/23/2014	AF_REPOSITORY rinn...		Edit View
□	getBrandAndModelForS...	SERVICE_CALL	1	SUBMITTED	12/17/2014	AF_REPOSITORY rinn...		Edit View
□	test12	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Extra Version A-AGLC L...	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Version A-AGLC Licens...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	NewVersion A-AGLC Lic...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Version B-AGLC Licens...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View

From the Home page, you can:

- Browse categories to find process objects in Object Library.
- Search across categories to find process objects by using Global Search.
- Create new processes, subprocesses, service calls, or context variables.
- Select process objects in the table and use the Cut and Paste buttons to reorganize process objects in the Object Library categories.
- Tag process objects with your own notes and keywords prior to submitting them for compilation and promotion.
- View process object version history by way of a version indicator link.

- Select actions in the table: view, edit, or compile a process. Clicking the Compile link creates a promotion record on the Promotion page, from which a fully validated and compiled top-level process and all its dependent objects (subprocesses, variables, service calls) can be promoted, which in turn creates an Information Manager document for the top-level process.

PROMOTE TAB

The Promote tab is visible to users whose administrator has given the privileges necessary to promote and publish AnswerFlow processes.

For more information, see “Promotions Page”.

CONFIGURATION TAB

The Configuration tab is visible only to users who have administrator privileges in AnswerFlow.

For more information, see “Configurations Page”.

DATA MIGRATION TAB

The Data Migration tab is visible to users who have the privilege to migrate data from an 8.5.x version of AnswerFlow to AnswerFlow 8.6.

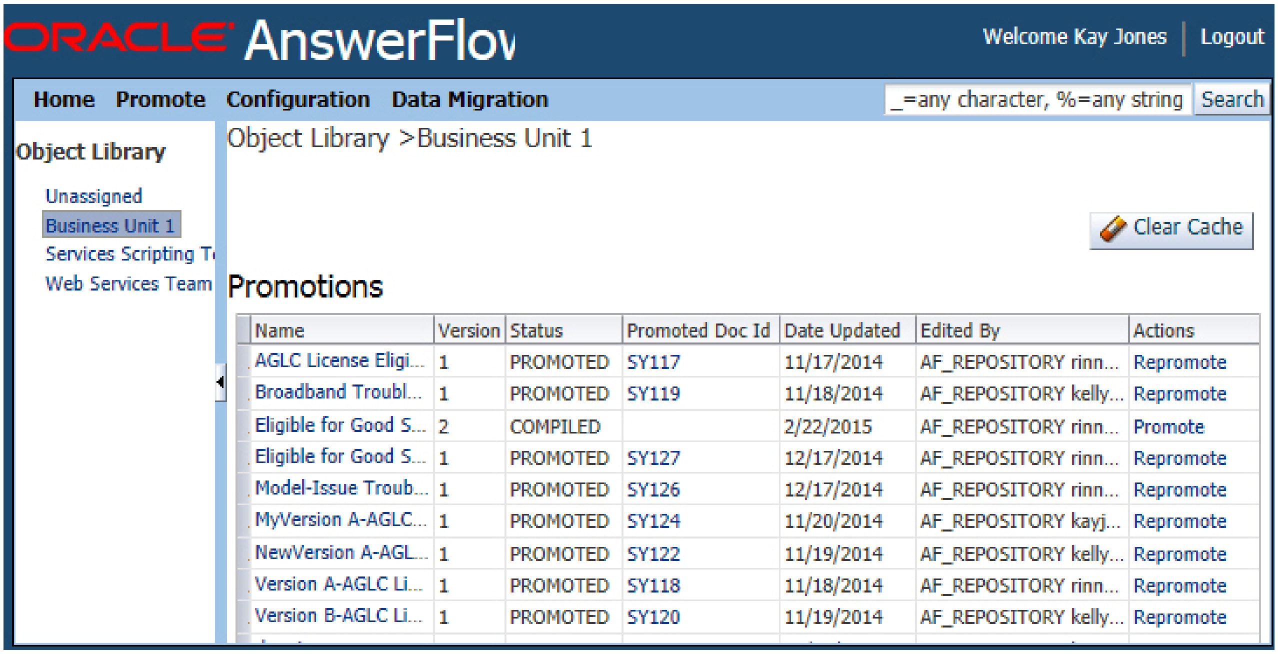
All AnswerFlow users use these categories to organize process objects. Click the arrow to expand any subcategories.

For more information, see “Data Migration Page”.

Promotions Page

The Promotions page lists all process flows that have been compiled and are ready for promotion and the process flows that have already been promoted.

On the Promotions page, the global header and the Object Library are the same as on the Home page. What is different is the information in the Promotions table, and different action buttons.

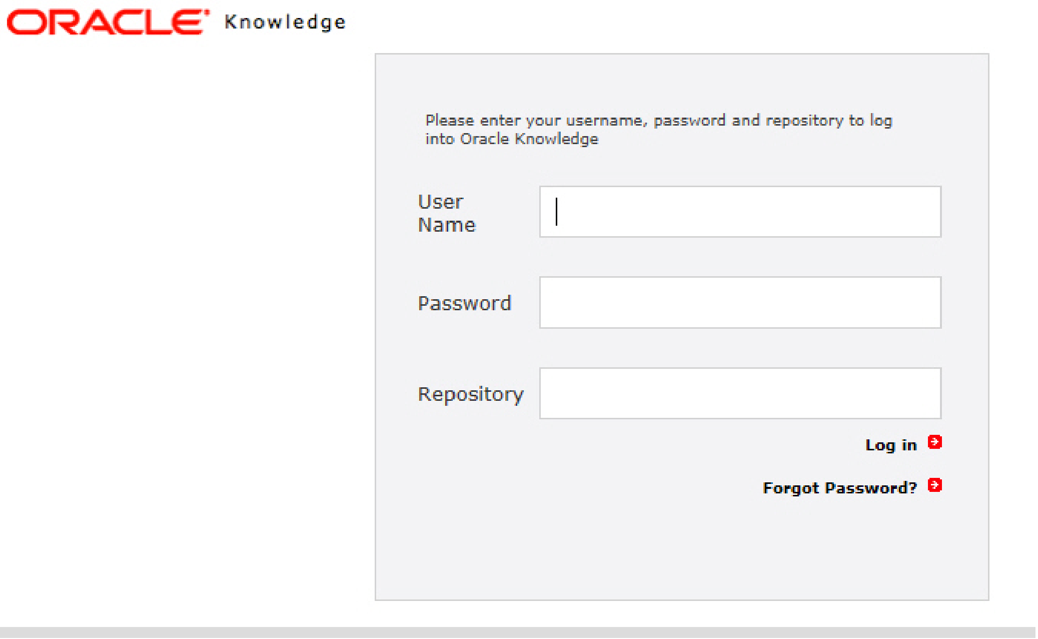


Name	Version	Status	Promoted Doc Id	Date Updated	Edited By	Actions
AGLC License Eligi...	1	PROMOTED	SY117	11/17/2014	AF_REPOSITORY rinn...	Repromote
Broadband Troubl...	1	PROMOTED	SY119	11/18/2014	AF_REPOSITORY kelly...	Repromote
Eligible for Good S...	2	COMPILED		2/22/2015	AF_REPOSITORY rinn...	Promote
Eligible for Good S...	1	PROMOTED	SY127	12/17/2014	AF_REPOSITORY rinn...	Repromote
Model-Issue Troub...	1	PROMOTED	SY126	12/17/2014	AF_REPOSITORY rinn...	Repromote
MyVersion A-AGLC...	1	PROMOTED	SY124	11/20/2014	AF_REPOSITORY kayj...	Repromote
NewVersion A-AGL...	1	PROMOTED	SY122	11/19/2014	AF_REPOSITORY kelly...	Repromote
Version A-AGLC Li...	1	PROMOTED	SY118	11/18/2014	AF_REPOSITORY rinn...	Repromote
Version B-AGLC Li...	1	PROMOTED	SY120	11/19/2014	AF_REPOSITORY kelly...	Repromote

In the Promotions table, you can click a process object in the Name column and see its details, most of which are presented in the table. The ID is noted in the detail popup, and may be helpful for troubleshooting.

On the Promotions page, users can:

- Promote compiled processes: This action pushes the compiled process to Information Manager as compressed data in a document awaiting publication.
- Re-promote processes: If you promote multiple versions of a process and then want to revert to a previous version, you can re-promote the previous version.
- Clear the cache: Click this button when you promote a new version of a process flow to confirm that the SampleUI dashboard is displaying the published version of a process under the Published tab.
- View the process within Information Manager: You publish processes by using Information Manager's content management functionality. After you publish a process in Information Manager, the process is published to the SampleUI dashboard, ready for end-user integration and for end users to use. Each promoted process has a Promoted Doc Id link that allows you to view the promoted process as it is rendered in Information Manager. When you click a promoted Doc Id link, the Information Manager login screen appears, as shown in the following figure.



ORACLE Knowledge

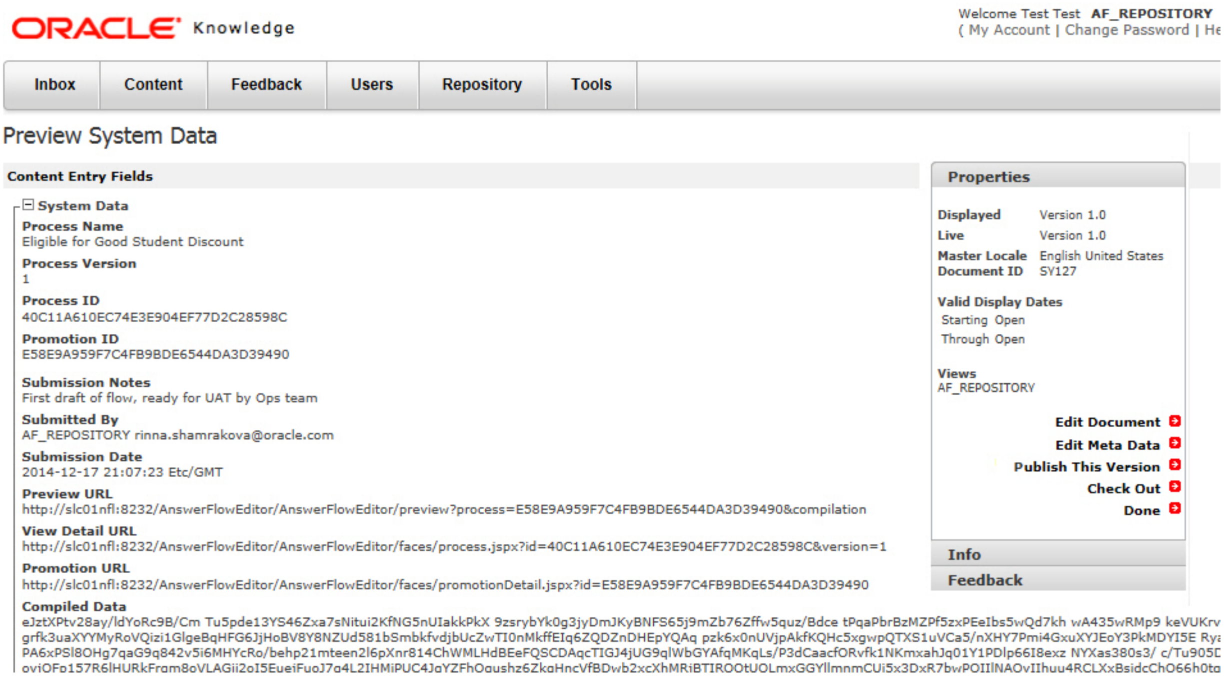
Please enter your username, password and repository to log into Oracle Knowledge

User Name

Password

Repository

After you sign in by using your Information Manager user name and password, you see a process within Information Manager similar to the following figure.



ORACLE Knowledge

Welcome Test Test AF_REPOSITORY
(My Account | Change Password | He

Preview System Data

Content Entry Fields

System Data

Process Name
Eligible for Good Student Discount

Process Version
1

Process ID
40C11A610EC74E3E904EF77D2C28598C

Promotion ID
E58E9A959F7C4FB9BDE6544DA3D39490

Submission Notes
First draft of flow, ready for UAT by Ops team

Submitted By
AF_REPOSITORY rinna.shamrakova@oracle.com

Submission Date
2014-12-17 21:07:23 Etc/GMT

Preview URL
http://slc01nfl:8232/AnswerFlowEditor/AnswerFlowEditor/preview?process=E58E9A959F7C4FB9BDE6544DA3D39490&compilation

View Detail URL
http://slc01nfl:8232/AnswerFlowEditor/AnswerFlowEditor/faces/process.jspx?id=40C11A610EC74E3E904EF77D2C28598C&version=1

Promotion URL
http://slc01nfl:8232/AnswerFlowEditor/AnswerFlowEditor/faces/promotionDetail.jspx?id=E58E9A959F7C4FB9BDE6544DA3D39490

Compiled Data

eJztXPtv28ay/ldYoRc9B/Cm Tu5pde13YS46Zxa7sNitui2KfNG5nUIakkPkX 9zsrybYk0g3jyDmJKyBNFS65j9mZb76Zffw5quz/Bdce tPqaPbrBzMZPf5zxPEeIbs5wQd7kh wA435wRMp9 keVUKrv
grfk3uaXYYMyRoVQizi1GlgeBqHFG6JjHoBV8Y8NZUd581bSmbkfvdjbUcZwTI0nMkffEIq6ZQDZnDHEpYQAq pzk6x0nUVjpAkfKQHc5xgwpQTXS1uVCa5/nXHY7Pmi4GxuXYJEoY3PkMDYI5E Rya
PA6xPSl8OHg7qaG9q842v5i6MHYcRo/behp21mteen2l6pXnr814ChWMLHdBEeFQSCDAqcTIGJ4jUG9qlWbGYAfqMKqLs/P3dCaacfORvfk1NKmxahJq01Y1PDlp66I8exz NYXas380s3/ c/Tu905D
oviOFp157R6lHURkFrqm8oVLAGij2oI5EueiFuoJ7q4L2IHMiPUC4JqYZFhOqushz6ZkqHncVfBDwb2xcXhMRiBTIROOtUOLmxGGYllmnmCUi5x3DxR7bwPOIIlNAOvIIhuu4RCLXxBsidcChO66h0tq

Properties

Displayed Version 1.0
Live Version 1.0
Master Locale English United States
Document ID SY127

Valid Display Dates
Starting Open
Through Open

Views
AF_REPOSITORY

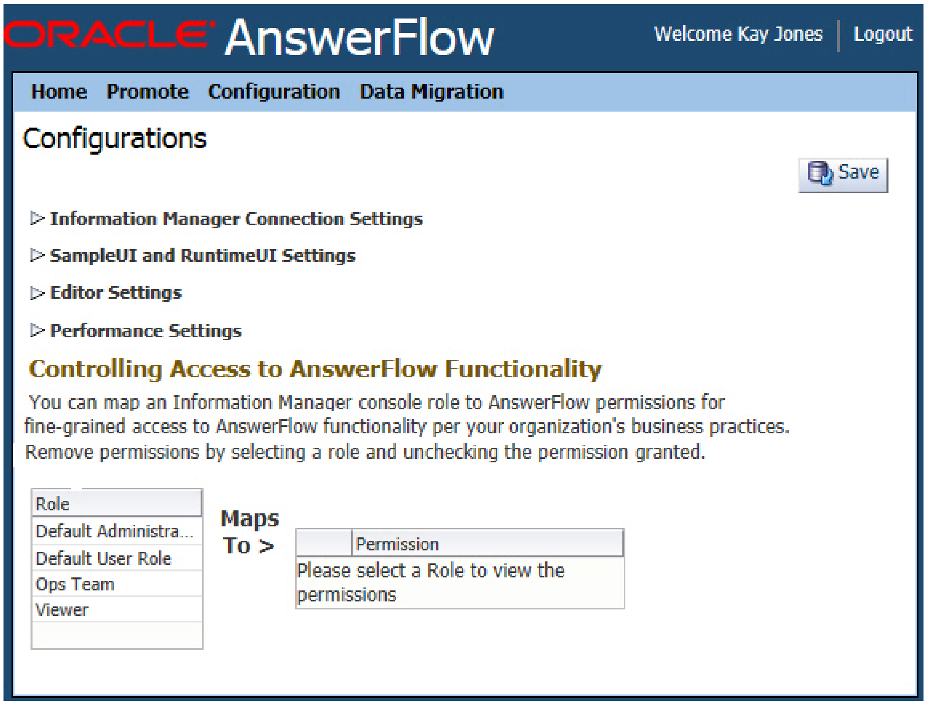
The previous figure shows a promoted process published in the AnswerFlow channel created for use with AnswerFlow in Information Manager. The Properties pane includes the Unpublish This Version link.

The AnswerFlow Channel in this example is called System Data, and it has a schema with one node called System Data that was defined before the installation and initial configuration of AnswerFlow.

Note: This node and its attributes cannot be edited; they must remain as defined in the initial installation for AnswerFlow and Information Manager to work together in the process flow storage and publishing processes.

Configurations Page

AnswerFlow application administrators use the Configurations page to maintain AnswerFlow settings and configurations. You can manage access control to AnswerFlow functionality by mapping Information Manager Security Roles to AnswerFlow privileges.


ORACLE AnswerFlow
Welcome Kay Jones | Logout
Home Promote Configuration Data Migration
Configurations
Save
▷ Information Manager Connection Settings
▷ SampleUI and RuntimeUI Settings
▷ Editor Settings
▷ Performance Settings

Controlling Access to AnswerFlow Functionality

You can map an Information Manager console role to AnswerFlow permissions for fine-grained access to AnswerFlow functionality per your organization's business practices. Remove permissions by selecting a role and unchecking the permission granted.

Role
Default Administra...
Default User Role
Ops Team
Viewer

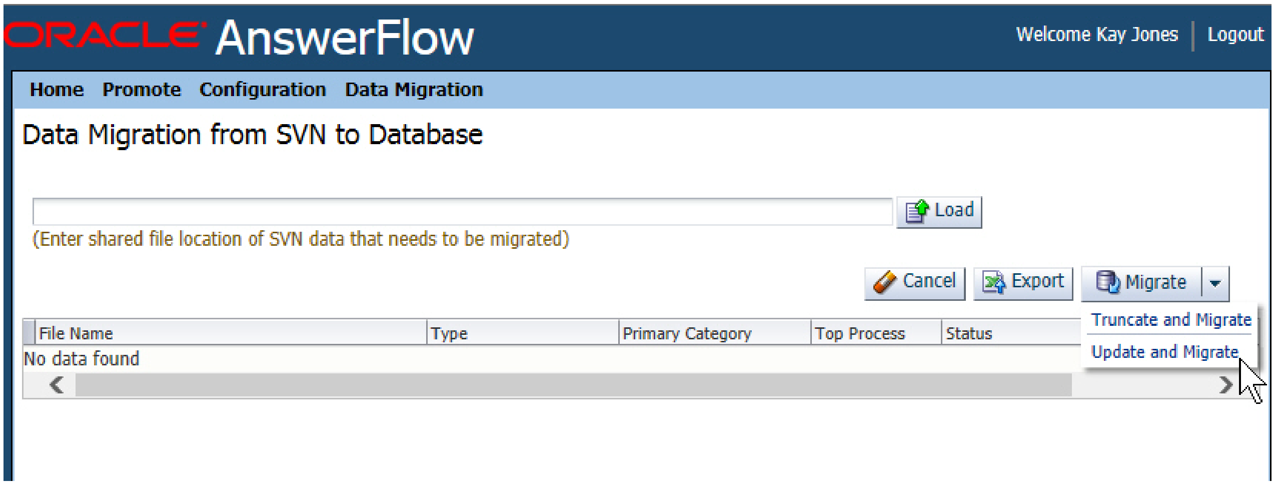
Maps To >

	Permission
Please select a Role to view the permissions	

For details on the Configuration page functionality, see “Configuring and Administering AnswerFlow”.

Data Migration Page

Administrators use the Data Migration page to load and transform AnswerFlow 8.5.x SVN object data for migration to the AnswerFlow 8.6 database.



Data Migration from SVN to Database

(Enter shared file location of SVN data that needs to be migrated)

File Name	Type	Primary Category	Top Process	Status
No data found				

For details on the functionality of the Data Migration page, see “Configuring and Administering AnswerFlow”.

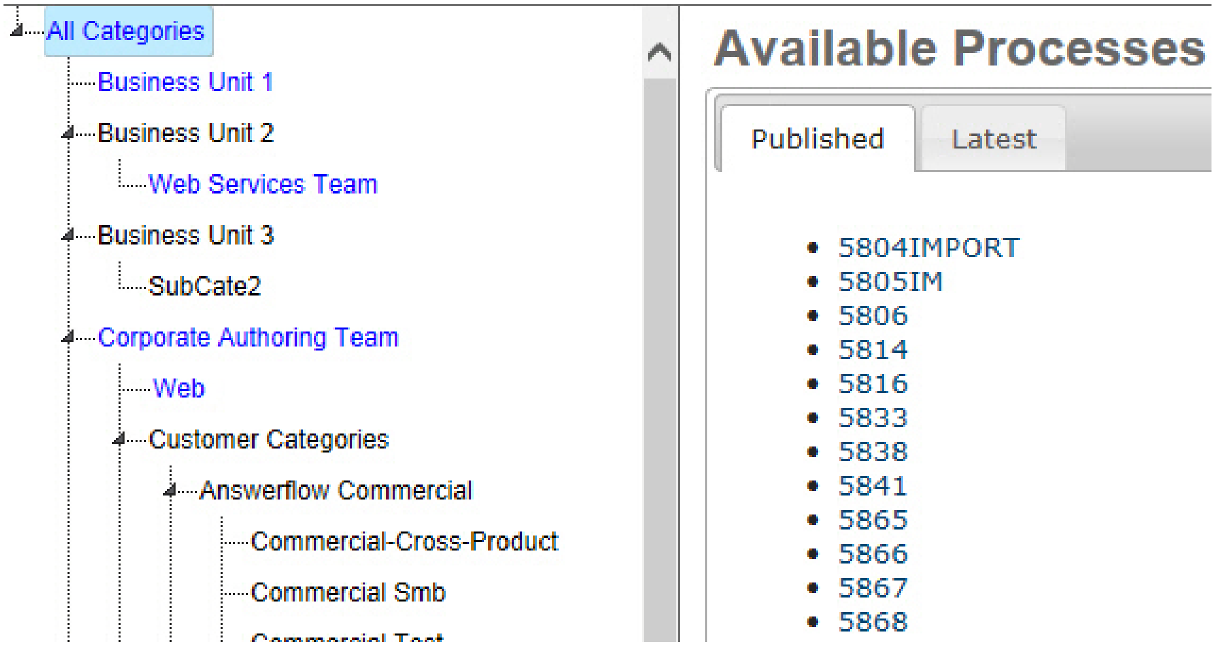
Using the AnswerFlow SampleUI Dashboard

This section describes how to navigate to and within the AnswerFlow SampleUI Dashboard.

To use the SampleUI Dashboard:

1 To access the Dashboard page, open a browser window and type the following URL:

```
http://<servername>:8233/SampleUI/dashboard
```



By default, the All Categories node is selected, so the Published tab lists all published process flows. The categories you see in SampleUI are those visible to the single Information Manager console user set up for SampleUI. This user was initially specified for SampleUI at installation, but can be changed later. For more information on which categories are visible to a user, see Category Assignments in About AnswerFlow Components.

Tip: If you have several flows, navigate to a specific category to more easily view smaller lists of processes.

2 Click the **Latest** tab to view processes that have been promoted and are the most recent version, but might not be published, if an earlier version of that same process is still published.

Note: You can use promoted process flows in the Latest tab for User Acceptance Testing before final publication and before making them available for integration with end-user web applications.

CHAPTER 2

Configuring and Administering AnswerFlow

Audience for This Section

This section focuses on the needs of an AnswerFlow application administrator who is responsible for the ongoing configuration and administration of the AnswerFlow Editor Web Application and the AnswerFlow SampleUI Dashboard. The administrator confirms that AnswerFlow is configured as needed to work properly with Information Manager and Oracle Knowledge Analytics.

This section helps administrators:

- Enable AnswerFlow to work with Information Manager
- Enable AnswerFlow to work with Oracle Knowledge Analytics
- Perform basic operations tasks to keep Answer Flow working for its users

Note: This guide does not cover any installation details. For information on installing AnswerFlow 8.6 as part of a new Oracle Knowledge installation, see *Installing and Configuring Oracle Knowledge*. For information on upgrading AnswerFlow 8.5x to version 8.6, see *Upgrading Oracle Knowledge Applications*.

Enabling AnswerFlow to Work With Information Manager

To use Information Manager to store and publish process flows, you create an AnswerFlow-specific channel. This task is completed before installing AnswerFlow, but is given here for reference, in case you need to maintain or otherwise update this channel and its schema.

AnswerFlow Editor and SampleUI process browsing is enabled by Information Manager categories applied to the AnswerFlow specific channel, so there may be occasions to update the categories assigned to the AnswerFlow channel if you have a reason to change the AnswerFlow browsing information architecture.

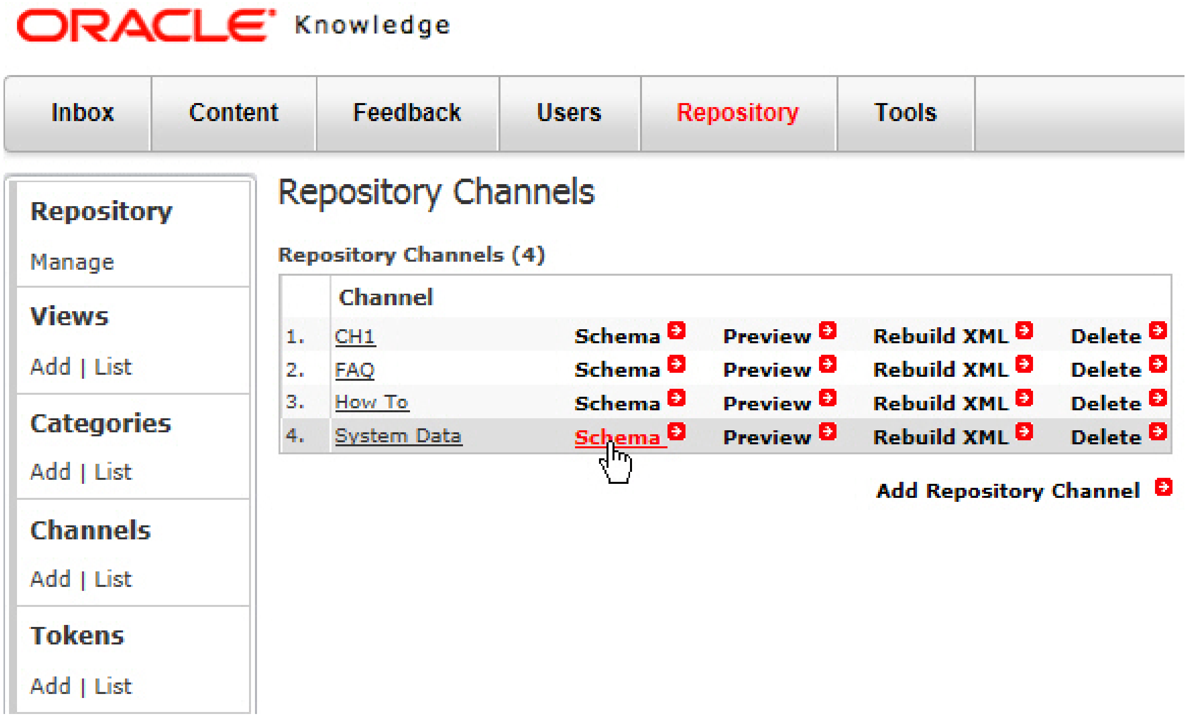
You must confirm that JGroups is enabled for AnswerFlow to work with Information Manager. For more information, see “Enabling JGroups”.

Working With the AnswerFlow Channel in Information Manager

Before installation, the person who installed AnswerFlow in your Oracle Knowledge instance created an AnswerFlow channel. This channel is the dedicated container for AnswerFlow processes after they are promoted from the AnswerFlow Editor Web Application. Instructions for creating this channel are in the *Oracle Knowledge Installation Guide*.

To edit the channel schema:

1. Sign in to Information Manager as an administrator to open the Preview page.
2. Click the **Repository** tab to open the Repository Management page.
3. In the left navigation pane, click the **Channels List** link to open the Repository Channels list.
4. Find your AnswerFlow Channel among the channels listed. In this example, it is a channel called System Data.
5. Click the **Schema** link for your AnswerFlow Channel, as shown in the following figure.

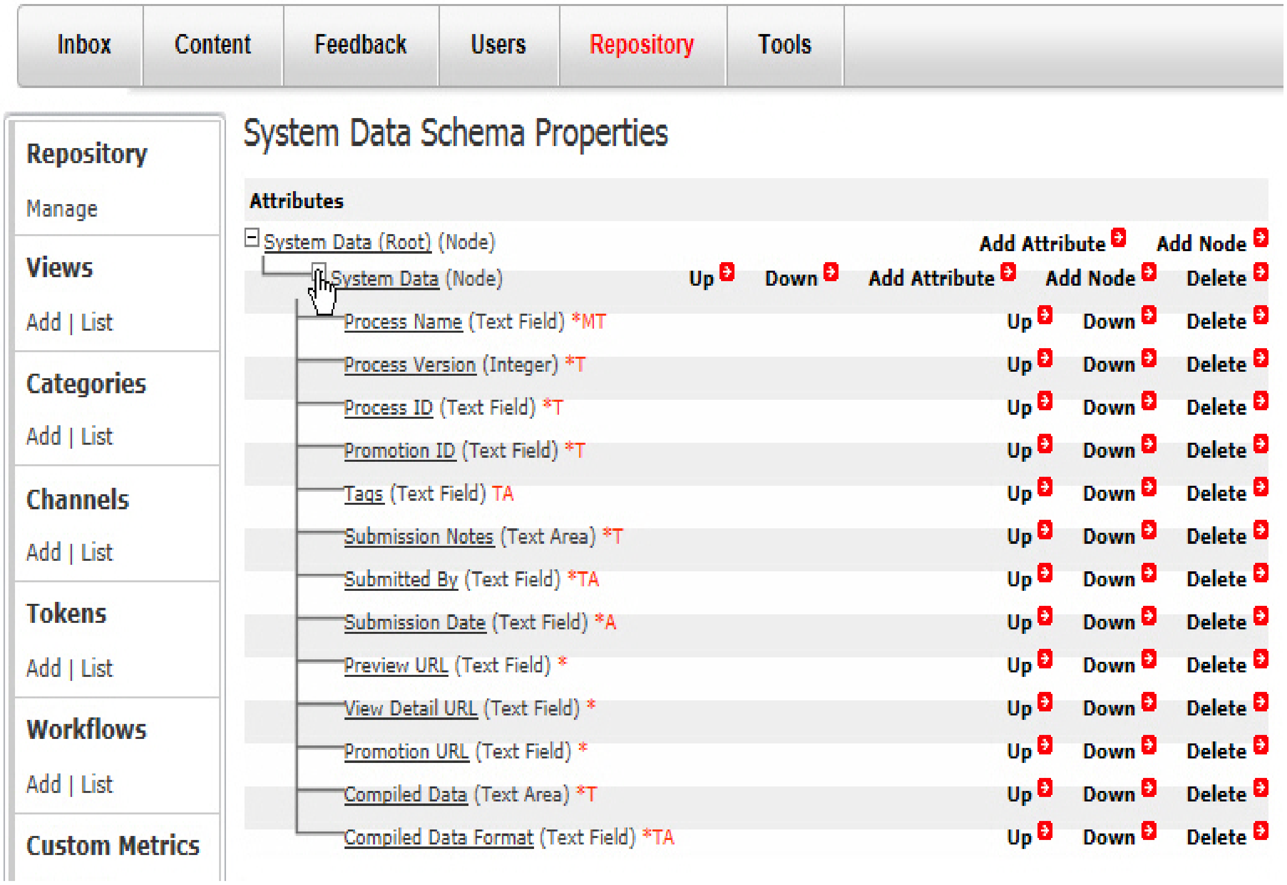


Repository Channels

Repository Channels (4)

	Channel				
1.	CH1	Schema	Preview	Rebuild XML	Delete
2.	FAQ	Schema	Preview	Rebuild XML	Delete
3.	How To	Schema	Preview	Rebuild XML	Delete
4.	System Data	Schema	Preview	Rebuild XML	Delete

The attribute node System Data is defined. To make the Information Manager document wrap the AnswerFlow processes effectively, you must define this attribute node.



System Data Schema Properties

Attributes

System Data (Root) (Node)				Add Attribute	Add Node
System Data (Node)	Up	Down	Add Attribute	Add Node	Delete
Process Name (Text Field) *MT			Up	Down	Delete
Process Version (Integer) *T			Up	Down	Delete
Process ID (Text Field) *T			Up	Down	Delete
Promotion ID (Text Field) *T			Up	Down	Delete
Tags (Text Field) TA			Up	Down	Delete
Submission Notes (Text Area) *T			Up	Down	Delete
Submitted By (Text Field) *TA			Up	Down	Delete
Submission Date (Text Field) *A			Up	Down	Delete
Preview URL (Text Field) *			Up	Down	Delete
View Detail URL (Text Field) *			Up	Down	Delete
Promotion URL (Text Field) *			Up	Down	Delete
Compiled Data (Text Area) *T			Up	Down	Delete
Compiled Data Format (Text Field) *TA			Up	Down	Delete

If you do not edit or change the System Data node, you can add other attributes to the channel. For example, you may want to add more content entry fields to this channel, such as a document title and abstract or summary, to aid in presenting search results (if you decide to crawl this AnswerFlow channel as an internal collection and find process flows in Semantic Search results). If you are not a frequent administrative user of Information Manager, you may want to consult with your organization's Information Manager administrator for guidance on editing the AnswerFlow channel.

The following table lists the values of the fields required for the System Data node.

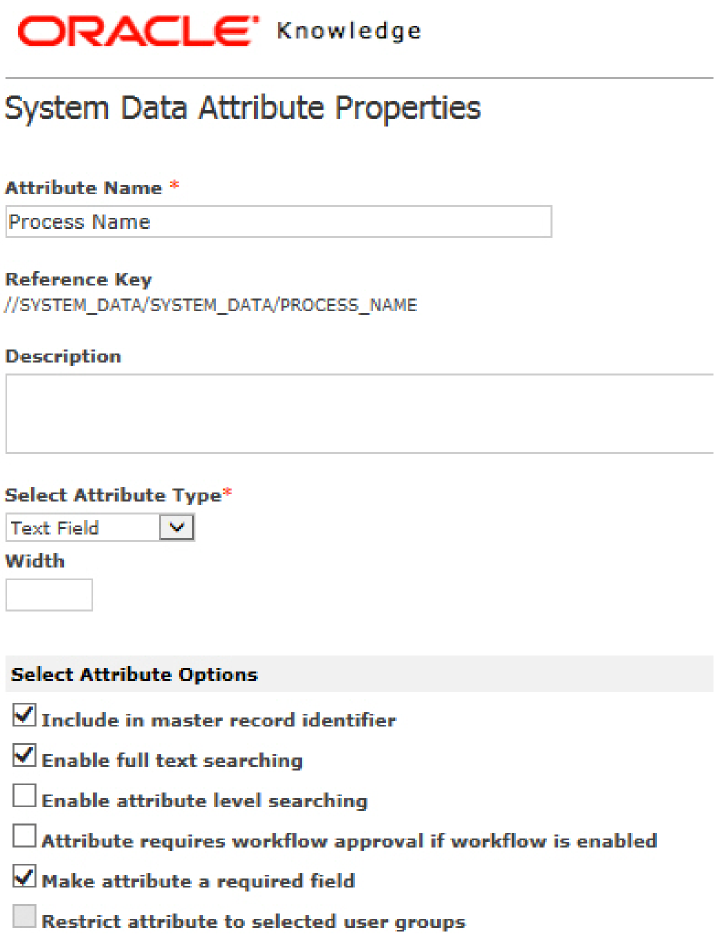
Important! Do not edit the System Data node and its attributes after they have been created. It is critical that the field types and reference keys are exactly as specified. To interface with AnswerFlow, you must define the nodes only as noted in the following table.

Nodes and Attributes	Field Type	Reference Key Name
-Root		
--System Data	(This is a non-repeating node.)	SYSTEM_DATA
---Process Name	*MT - Text Field	PROCESS_NAME
---Process Version	*T - Integer	PROCESS_VERSION
---Process ID	*T - Text Field	PROCESS_ID
---Promotion ID	*T - Text Field	PROMOTION_ID
---Tags	TA - Text Field	TAGS
---Submission Notes	*T - Text Area	SUBMISSION_NOTES

---Submitted By	*TA - Text Field	SUBMITTED_BY
---Submission Date	*A - Date Time	SUBMISSION_DATE
---Preview URL	* - Text Field	PREVIEW_URL
---View Detail URL	* - Text Field	VIEW_DETAIL_URL
---Promotion URL	* - Text Field	PROMOTION_URL
---Compiled Data	* - Text Area	COMPILED_DATA
---Compiled Data Format	T - Text Field	COMPILED_DATA_FORMAT

- * - Required Field
- M - Master Identifier
- T - Included in full text search
- A - Available in attribute search

The following figure shows the Attribute Properties page for the Process Name attribute of the System Data node.



ORACLE Knowledge

System Data Attribute Properties

Attribute Name *

Process Name

Reference Key

//SYSTEM_DATA/SYSTEM_DATA/PROCESS_NAME

Description

Select Attribute Type*

Text Field

Width

Select Attribute Options

☑ Include in master record identifier

☑ Enable full text searching

☐ Enable attribute level searching

☐ Attribute requires workflow approval if workflow is enabled

☑ Make attribute a required field

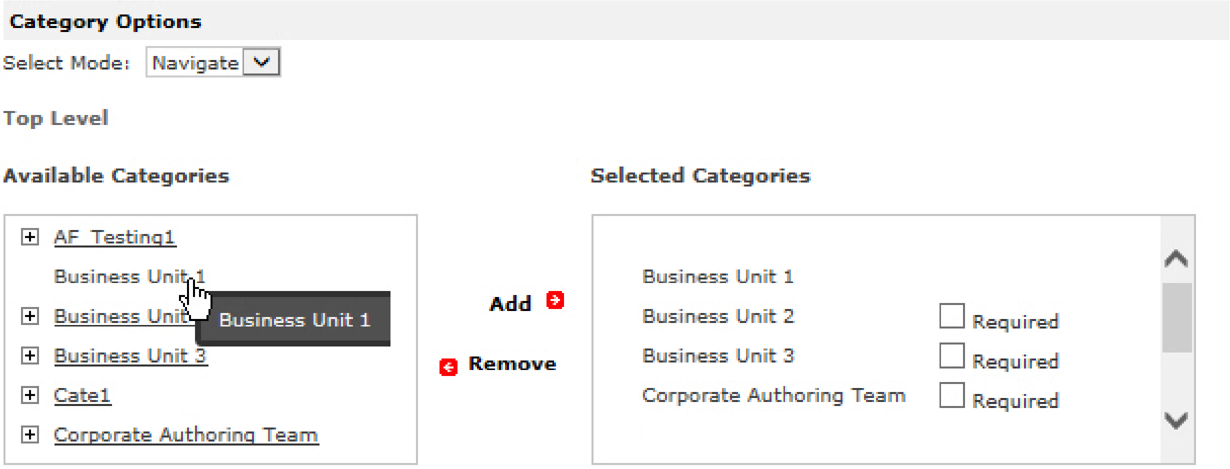
☐ Restrict attribute to selected user groups

Applying Categories to the AnswerFlow Channel

To update the AnswerFlow Editor Web Application navigation tree, use the following procedure to update your channel categories:

1. Sign in to Information Manager as an administrator.
2. Click the **Repository** tab.
3. On the **Channels** menu, click **List**.
4. Click the AnswerFlow channel name to open the Channel properties page.

   The Channel properties page is where you add and edit categories, assign user roles access, and assign a workflow to the AnswerFlow channel.
5. Scroll to the **Available Categories** pane.



Category Options

Select Mode: Navigate

Top Level

Available Categories

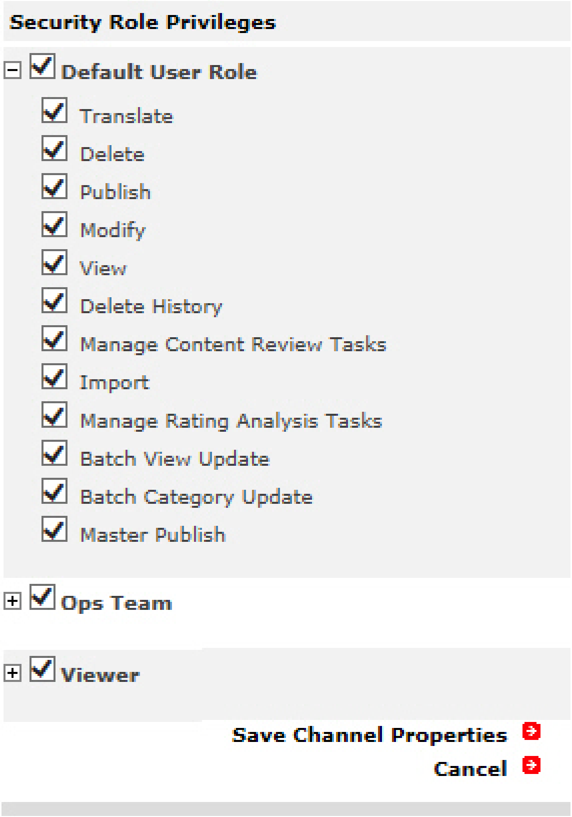
⊞ AF_Testing1
Business Unit 1
⊞ Business Unit 2
⊞ Business Unit 3
⊞ Cate1
⊞ Corporate Authoring Team

Selected Categories

Business Unit 1
Business Unit 2 □ Required
Business Unit 3 □ Required
Corporate Authoring Team □ Required

6. Use the **Add** and **Remove** arrows to expose selected categories in this channel. You can also add new categories to this list.

   Note: Consult with your Information Manager administrator before adding new categories to the Information Manager category list.
7. Scroll to the **Security Role Privileges** pane and confirm that all the roles you use for AnswerFlow users are checked, otherwise this channel will not be visible to your AnswerFlow Editor Web Application users.



Security Role Privileges

⊟ ☑ **Default User Role**

- ☑ Translate
- ☑ Delete
- ☑ Publish
- ☑ Modify
- ☑ View
- ☑ Delete History
- ☑ Manage Content Review Tasks
- ☑ Import
- ☑ Manage Rating Analysis Tasks
- ☑ Batch View Update
- ☑ Batch Category Update
- ☑ Master Publish

⊞ ☑ **Ops Team**

⊞ ☑ **Viewer**

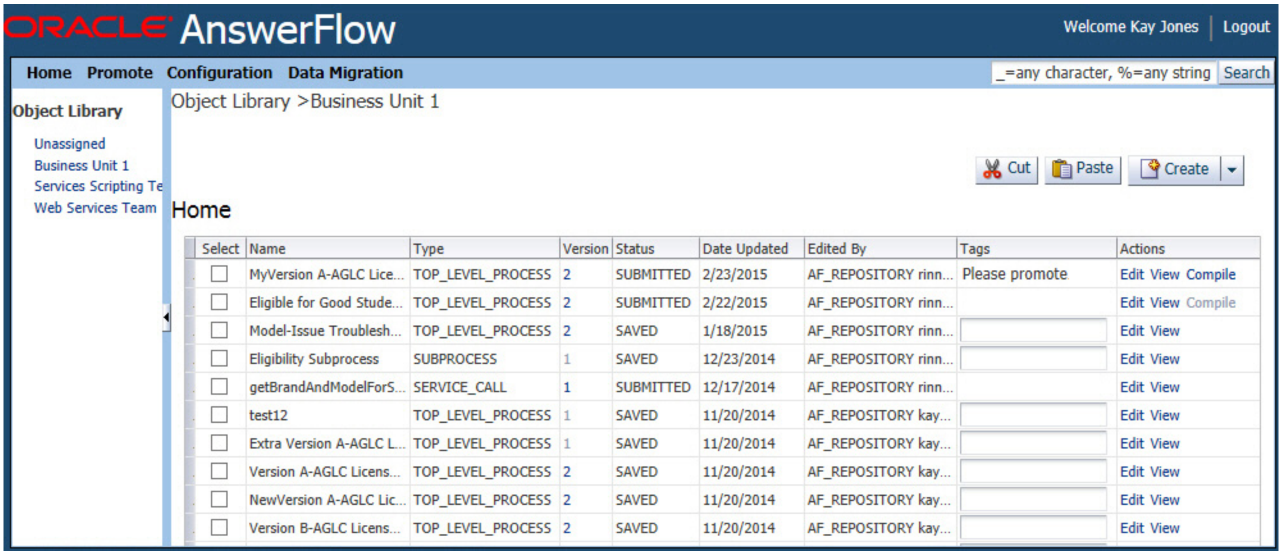
8 When finished, click the **Save Channel Properties** link to return to the Repository Channels page.

9 Click the **Users** tab.

10 On the **Console User** menu, click **List**.

11 Click on each Console User who will use AnswerFlow to review the categories assigned to that user. Confirm that your AnswerFlow users have the categories they need to view process flows in AnswerFlow. Your AnswerFlow administrator should have all AnswerFlow categories available, but you could separate AnswerFlow users by category.

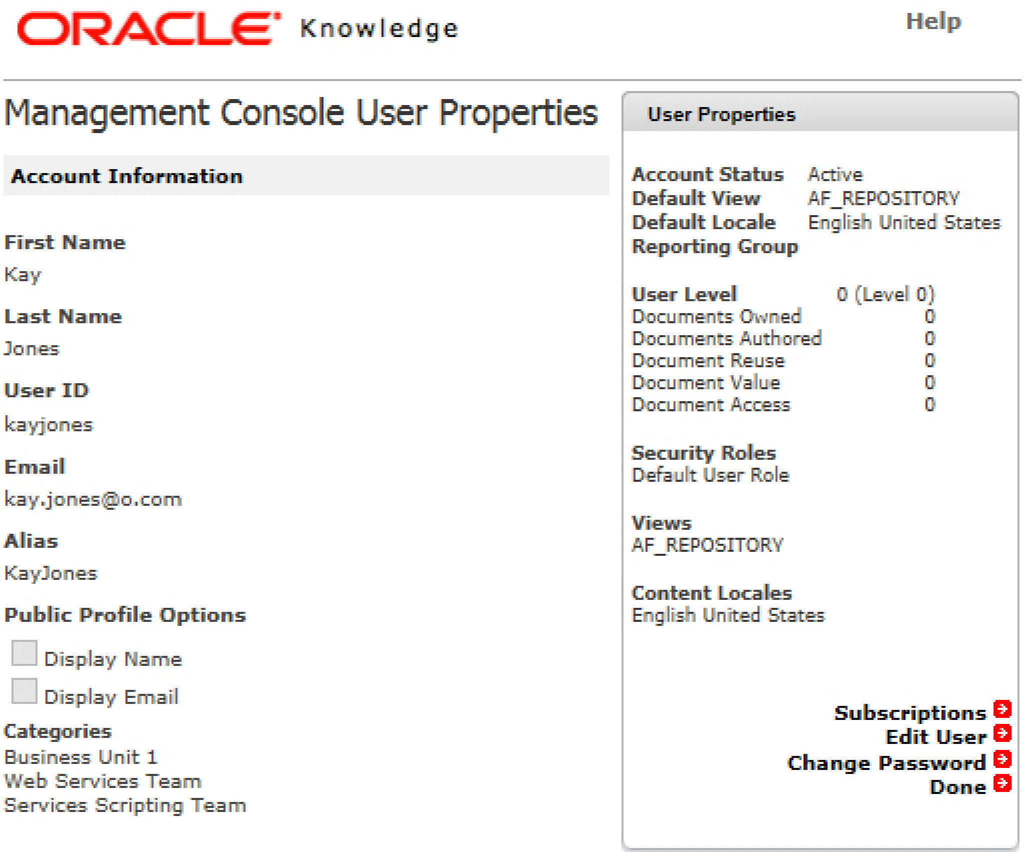
Example: In the following figure, Console user Kay Jones is an AnswerFlow user who can see process objects in AnswerFlow under the categories Business Unit 1, Services Scripting Team, and Web Services Team. She cannot navigate to process objects for Business Unit 2 or 3, because those categories are not assigned to her.



Home

Select	Name	Type	Version	Status	Date Updated	Edited By	Tags	Actions
□	MyVersion A-AGLC Lice...	TOP_LEVEL_PROCESS	2	SUBMITTED	2/23/2015	AF_REPOSITORY rinn...	Please promote	Edit View Compile
□	Eligible for Good Stude...	TOP_LEVEL_PROCESS	2	SUBMITTED	2/22/2015	AF_REPOSITORY rinn...		Edit View Compile
□	Model-Issue Troublesh...	TOP_LEVEL_PROCESS	2	SAVED	1/18/2015	AF_REPOSITORY rinn...		Edit View
□	Eligibility Subprocess	SUBPROCESS	1	SAVED	12/23/2014	AF_REPOSITORY rinn...		Edit View
□	getBrandAndModelForS...	SERVICE_CALL	1	SUBMITTED	12/17/2014	AF_REPOSITORY rinn...		Edit View
□	test12	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Extra Version A-AGLC L...	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Version A-AGLC Licens...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	NewVersion A-AGLC Lic...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Version B-AGLC Licens...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View

12 If the Security Roles section does not include the desired Information Manager Security Roles for this user, click the **Edit User** link to open the User Properties page, as shown in the following figure.



Management Console User Properties

Account Information

First Name
Kay

Last Name
Jones

User ID
kayjones

Email
kay.jones@o.com

Alias
KayJones

Public Profile Options

□ Display Name

□ Display Email

Categories
Business Unit 1
Web Services Team
Services Scripting Team

User Properties

Account Status	Active
Default View	AF_REPOSITORY
Default Locale	English United States
Reporting Group	

User Level	0 (Level 0)
Documents Owned	0
Documents Authored	0
Document Reuse	0
Document Value	0
Document Access	0

Security Roles
Default User Role

Views
AF_REPOSITORY

Content Locales
English United States

13 Add the missing roles for the permissions that the user should have. Depending on your Information Manager setup, you may need users to have certain Security Roles to perform actions involving documents, such as promoting a process to a document.

Connecting SampleUI to Information Manager

The installation specified a single user for SampleUI access. An administrator can edit the user credentials on the AnswerFlow Editor Configuration page.

You can also use the following Java service to customize the SampleUI and Information Manager connections. You can connect SampleUI to Information Manager by modifying the authorization settings in the sample user name service (`SampleUserNameService.java`).

To modify the sample user name service:

1 Edit the file `SampleUserNameService.java`, located at:

```
${ANSWERFLOW_INSTALL_DIRECTORY}/AnswerFlow/Sample/SampleUI/src/main/java/com/
inquira/pmeditor/security/SampleUserNameService.java
```

2 Edit the following methods as appropriate for your environment:

Method	Returns
getCurrentUserName	Name of a user who has permission to access the configured Information Manager instance. This method accepts credentials as a parameter.
getUserName	Name of a user who has permission to access the configured Information Manager instance.
getCurrentPassword	User's password. This method accepts credentials as a parameter.
getPassword	User's password.
getCurrentRepository	Reference key of the Information Manager repository that contains the documents. This method accepts credentials as a parameter.
getRepositoryReferenceKey	Reference key of the Information Manager repository that contains the documents.

Enabling JGroups

You must enable JGroups for AnswerFlow to use Information Manager for process storage and publishing.

To enable JGroups:

1 Open the following files:

```
"$IM_INSTALLATION_DIRECTORY/InfoManager/config/IMADMIN/application.properties
"$IM_INSTALLATION_DIRECTORY/InfoManager/config/IMWS/application.properties
```

Note: If `$IM_INSTALLATION_DIRECTORY/InfoManager/config/$ANSWERFLOW_REPOSITORY/application.properties` does not exist, then do not create it.

2 Locate the following lines, which may or may not be commented out:

```
#jgroups.eofsynchronizer.enabled=true
#jgroups.eofsynchronizer.group.identifier=InquiraEOFGroup
```

3 If these lines are not commented out, then JGroups is already enabled. If these lines are commented out, continue to step 4.

4 Remove the comments (remove the # at the beginning of each line).

5 Make sure that both files have the same value for `jgroups.eofsynchronizer.group.identifier`

6 Save the files.

Administering the AnswerFlow Editor Web Application

AnswerFlow application administrators can find procedures in this section for:

- Managing AnswerFlow properties and configuration settings
- Managing access control to AnswerFlow functionality
- Setting up a Memcached server to improve SampleUI performance
- Migrating AnswerFlow data from an earlier version to an 8.6 installation
- Starting and stopping the AnswerFlow Editor web application and the RuntimeUI application
- Enabling AnswerFlow to work with Oracle Knowledge Analytics

Managing AnswerFlow Properties and Access

You manage AnswerFlow properties and configurations on the Configurations page. Access to this page should be limited to AnswerFlow application administrators. Administrators use the Configurations page to edit Configuration properties, which were set at installation. Administrators can also map Information Manager security roles to specific AnswerFlow privileges here to control access to AnswerFlow functionality.

Managing AnswerFlow Properties

Four groups of configuration properties are listed below, followed by detailed tables describing the specific properties in each group.

- Information Manager connection properties
- SampleUI and RuntimeUI connection properties
- Editor properties
- Performance properties

Note: In the tables that follow, if a property is marked with a asterisk, for that value to take effect you must restart the AnswerFlow Runtime server after changing the property value.

The following tables describe the configuration properties listed in the AnswerFlow Editor Configuration tab.

Property	Description
`common.imwsUrl`	URL of Information Manager Web Services web application to use with AnswerFlow. Example: `http://hostname:8226/imws`
`common.imResourceUrl`	URL of Information Manager resources web application to use with AnswerFlow. Example: `http://hostname:8226/resources`

`common.imConsoleUrl`	URL of Information Manager Console web application to use with AnswerFlow, followed by the string:
	`/WebObjects/InfoManager.woa?contentid=`
	Example:
	`http://hostname:8226/InfoManager/WebObjects/InfoManager.woa?contentid=`

`common.credentials.repository`	Information Manager repository reference key that AnswerFlowEditor and SampleUI use for authentication.
`common.channel`	Reference key for the Information Manager channel used to determine the Information Manager categories for AnswerFlow objects and to store the promoted top-level process documents.

Property	Description
`sampleUi.path.js`	URL path to SampleUI JavaScript resources (Default: `/SampleUI/resources/js`)
`sampleUi.path.base`	URL path to SampleUI web application. (Default: `/SampleUI`)
`sampleUi.domainAndPort`	Schema, domain, and port for SampleUI. You can leave this property blank if the SampleUI web application has the same scheme, domain, and port as the web application that executes the SampleUI process-rendering JavaScript. For example, `http://hostname:8233 (Default: <blank>)` For more information on `domainAndPort`, see “Version Specific Configurations:”.
`sampleUi.path.css`	URL path to SampleUI CSS resources. (Default: `/SampleUI/resources/css`)
`sampleUi.path.jmvc`	Not used. (Default: `/SampleUI/resources/jmvc`)
`sampleUi.path.img`	Not used. (Default: `/SampleUI/resources/images`)
`sampleUi.credentials.username`	Information Manager user name that SampleUI uses for authentication.
`sampleUi.credentials.password`	Encrypted Information Manager password that SampleUI uses for authentication. If you want to change the password, enter it in plain text and click Save. AnswerFlow encrypts the password for you.

Table 1: Editor Properties

Property	Description
`editor.imageUrl`	Scheme, domain, and port of the AnswerFlowEditor web application used to serve AnswerFlowEditor images. It fetches images during process PDF export. This property can be left blank if the PDF export process should use the HTTP request to infer the image URL. Example: `http://hostname:8232` (Default: `<blank>`)
`editor.url`	URL of AnswerFlowEditor. Example: `http://hostname:8232/AnswerFlowEditor`
`editor.maximumNumberOfUndos`	The maximum number of times a user can undo in AnswerFlow Editor.

Property	Description
`common.maximumConcurrentlyExecut-ingProcessesPerSession`	Maximum number of executions of top-level processes that a single HTTP session can have active at the same time. This is usually relevant only if a user is executing processes in multiple browser tabs. Use this property for both AnswerFlowEditor (for Preview) and SampleUI. (Default: 24).
`common.memcachedServer`	Domain and port of a Memcached server, or blank if not using Memcached.

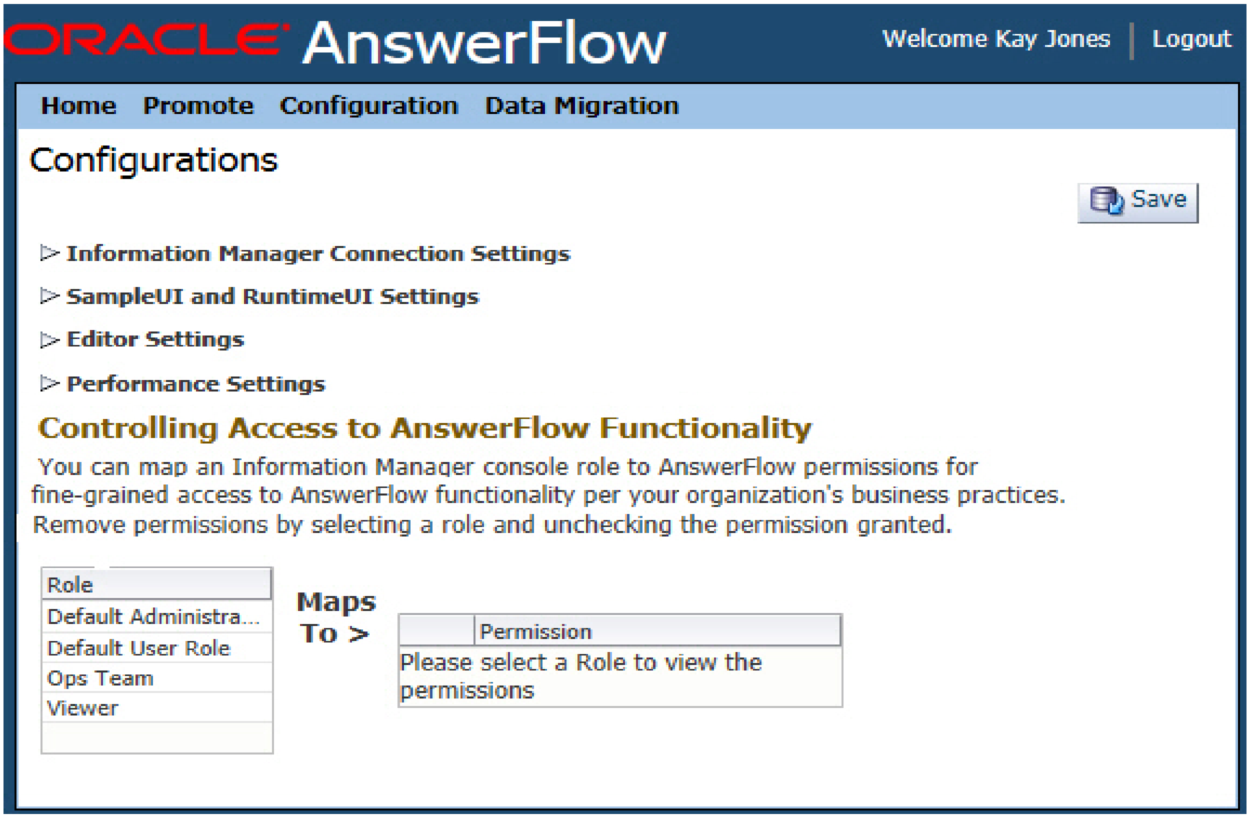
Managing AnswerFlow Access Control

AnswerFlow users are Information Manager console users who have access to the AnswerFlow channel that was created to store their processes. All Information Manager console users in the repository have authorization to use the AnswerFlow Editor.

You can fine-tune the access control to AnswerFlow functionality in the AnswerFlow Configuration page by mapping Information Manager Security Roles to AnswerFlow privileges.

To map Information Manager Security Roles to AnswerFlow privileges:

1. Sign in to AnswerFlow as an administrator to open the Home page.
2. Click the **Configuration** link to open the Configurations page.
3. Scroll to **Controlling Access to AnswerFlow Functionality**.


ORACLE AnswerFlow
Welcome Kay Jones | Logout
Home Promote Configuration Data Migration
Save

Configurations

▷ Information Manager Connection Settings

▷ SampleUI and RuntimeUI Settings

▷ Editor Settings

▷ Performance Settings

Controlling Access to AnswerFlow Functionality

You can map an Information Manager console role to AnswerFlow permissions for fine-grained access to AnswerFlow functionality per your organization's business practices. Remove permissions by selecting a role and unchecking the permission granted.

Role
Default Administra...
Default User Role
Ops Team
Viewer

Maps To >

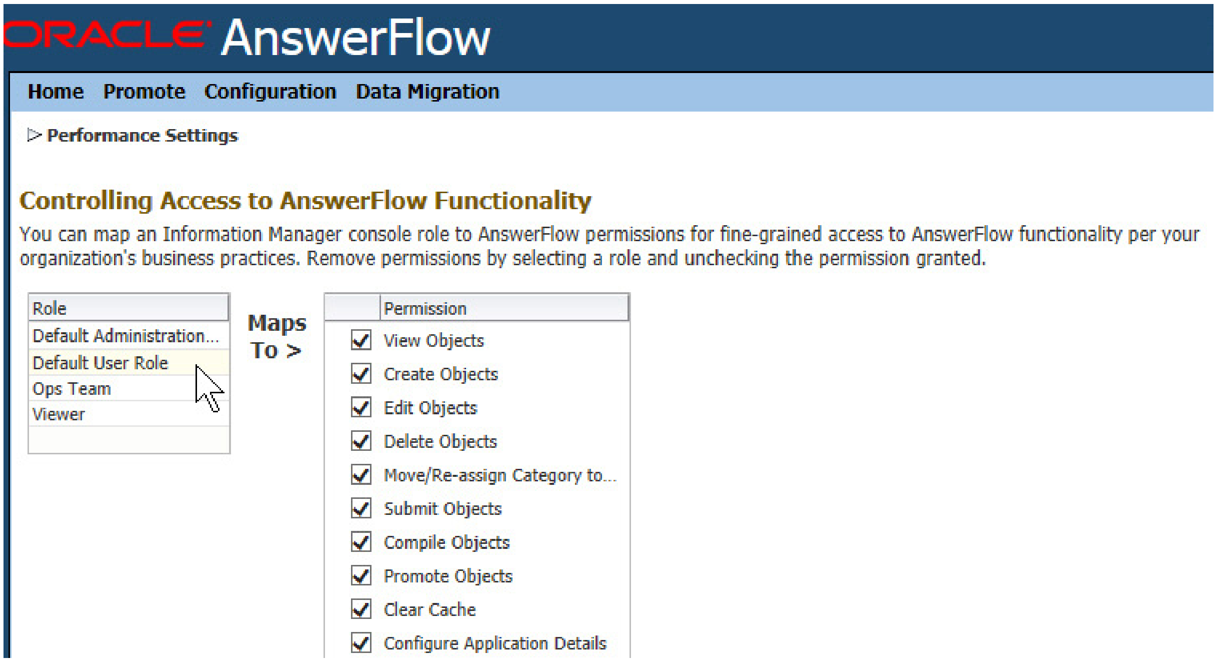
	Permission
Please select a Role to view the permissions	

The Role column lists all the Information Manager Security Roles that have been given access to the AnswerFlow channel.

If you are expecting to see a role that is not listed, you must edit User Console Role properties in Information Manager, confirming that the roles in question have access to the AnswerFlow Channel.

4 Select the name of a role in the **Role** column.

The Permissions column lists the corresponding AnswerFlow privileges, as in the following example figure.



Controlling Access to AnswerFlow Functionality

You can map an Information Manager console role to AnswerFlow permissions for fine-grained access to AnswerFlow functionality per your organization's business practices. Remove permissions by selecting a role and unchecking the permission granted.

Role
Default Administration...
Default User Role
Ops Team
Viewer

	Permission
☑	View Objects
☑	Create Objects
☑	Edit Objects
☑	Delete Objects
☑	Move/Re-assign Category to...
☑	Submit Objects
☑	Compile Objects
☑	Promote Objects
☑	Clear Cache
☑	Configure Application Details

5 In the **Permission** column, click the privileges that you want to map to the Information Manager security role that you selected.

6 Repeat step 4 and step 5 for all the roles that you want to map.

7 When you are finished mapping, click the **Save** button.

Role-privilege mapping is now in effect and users see only the functions to which their Information Manager Security Role is mapped.

Note: You cannot change the Default Administrator Role. By default, this role has all AnswerFlow privileges and is not editable in this interface.

Setting Up Memcached

Memcached is a distributed cache that reduces database loads, resulting in faster dynamic web applications. The main benefit of Memcached to AnswerFlow users is the ability to retrieve process flows from SampleUI quickly by storing them in a cache. This ability can be critical for organizations with a large number of users who attempt to access the same process flow (as wrapped by an Information Manager document) simultaneously.

Note: Installing Memcached is optional. You do not need to set up Memcached to use AnswerFlow, although using Memcached will improve the performance of the SampleUI. If you opt to use it, keep Memcached running whenever the SampleUI is running so that the SampleUI can cache data in the Memcached server.

To install Memcached:

1 On the server that you will run Memcached, download and extract the file `https://github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz.`

2 cd to the extracted `libevent-2.0.21-stable directory`.

3 Run `./configure --prefix=<prefix>`, and replace <prefix> with the directory prefix you wish to use for the libevent installation, such as `/usr/local`.

4 Run `make`.

5 Run `sudo make install` (the sudo in this command is only required if the current user does not have sufficient write permissions within the installation prefix).

6 Download and extract `https://github.com/memcached/memcached/archive/1.6.0-beta1.tar.gz`

7 cd to the extracted `Memcached-1.6.0-beta1` directory.

8 Run `autoconf`.

9 Run `./configure --prefix=<prefix>`, and replace <prefix> with the directory prefix you wish to use for the libevent installation, such as `/usr/local`.

10 Run `make`.

11 Run `sudo make install` (the sudo in this command is only required if the current user does not have sufficient write permissions within the installation prefix).

Memcached is installed.

12 For usage instructions, execute `man memcached`.

Migrating Data from Previous Versions of AnswerFlow

If you are upgrading from AnswerFlow 8.5.x to AnswerFlow 8.6, use the AnswerFlow data migration tool to migrate your 8.5.x AnswerFlow SVN repository to the 8.6 AnswerFlow database table.

Note: If your first version of Oracle Knowledge is 8.6, you can ignore this section, because you will not have any prior version AnswerFlow data to migrate.

This section includes instructions for creating a CSV file that provides the data migration tool with the needed information about your AnswerFlow 8.5.x SVN repository. The information enables the transformation of this data into the AnswerFlow 8.6 database table.

This section also includes instructions on running the data migration tool.

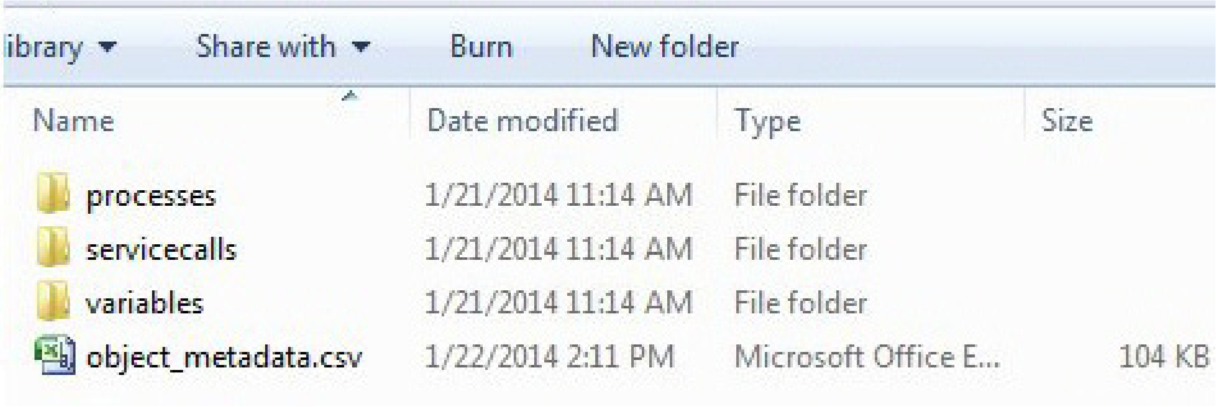
Important! If possible, run your migration process at a time when no one else is accessing the same table.

To migrate data:

1 Find the location of your AnswerFlow 8.5.x SVN objects. The likely location is:
`<AnswerFlow installation directory>/data/Editor/WORKAREA_DATA/PM_WorkingCopy/ProcessModel/development/ProcessModelConfigService`

The CSV file that you will create, named `object_metadata.csv`, will go into `ProcessModelConfigService`.

The folder has three subfolders: processes, servicecalls, and variables—the AnswerFlow objects to migrate. The processes folder holds both top-level processes and subprocesses to differentiate in the CSV file.

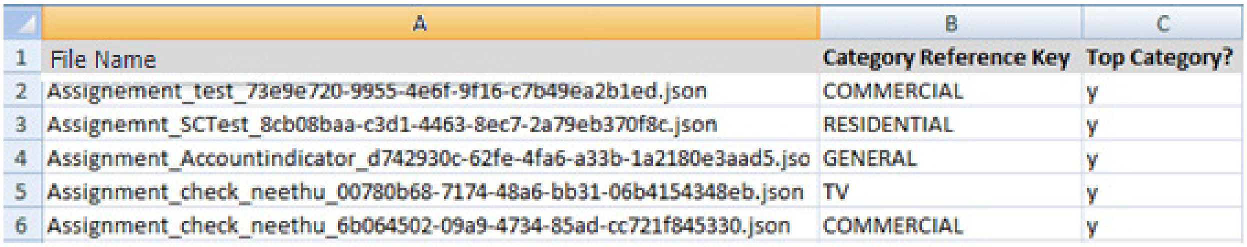


Name	Date modified	Type	Size
processes	1/21/2014 11:14 AM	File folder	
servicecalls	1/21/2014 11:14 AM	File folder	
variables	1/21/2014 11:14 AM	File folder	
object_metadata.csv	1/22/2014 2:11 PM	Microsoft Office E...	104 KB

2 Create the `object_metadata.csv` file to place in the `ProcessModelConfigService` folder.Open a spreadsheet program and create a CSV file with the following columns:

- File Name: Name of the JSON object files within the processes.
- Category Reference Key: Information Manager Category reference key to which you assign this object.
- Top Level Process?: Specify Yes or No with Y or N.

The following figure shows an example of an `object_metadata.csv` file.

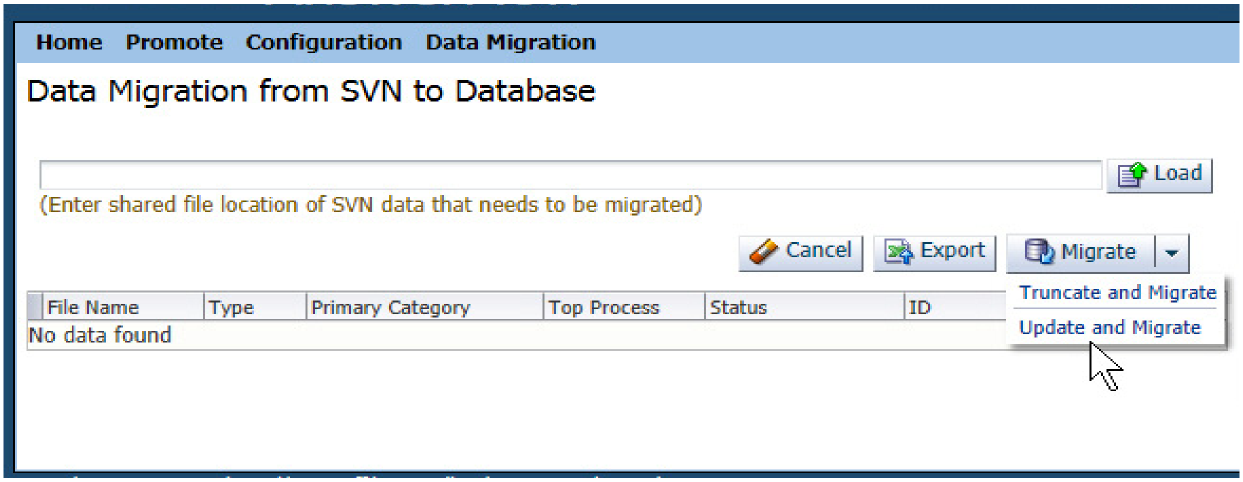


	A	B	C
1	File Name	Category Reference Key	Top Category?
2	Assignement_test_73e9e720-9955-4e6f-9f16-c7b49ea2b1ed.json	COMMERCIAL	y
3	Assignemnt_SCTest_8cb08baa-c3d1-4463-8ec7-2a79eb370f8c.json	RESIDENTIAL	y
4	Assignment_Accountindicator_d742930c-62fe-4fa6-a33b-1a2180e3aad5.jso	GENERAL	y
5	Assignment_check_neethu_00780b68-7174-48a6-bb31-06b4154348eb.json	TV	y
6	Assignment_check_neethu_6b064502-09a9-4734-85ad-cc721f845330.json	COMMERCIAL	y

> **Note:** The `object_metadata.csv` file sometimes faces a locking issue in which the user sees a message such as, “The system cannot find the specified file”, even though the file can be opened by Notepad or Excel. To fix this, rename the `object_metadata.csv` file to `object_metadata.csv.bak` and create an empty `object_metadata.csv` file.

3 After you prepare the CSV file and place it in the `ProcessModelConfigService` folder, sign in to AnswerFlow as an administrator.

4 On the menu bar click the **Data Migration** tab to open the Data Migration from SVN to Database page.



Data Migration from SVN to Database

(Enter shared file location of SVN data that needs to be migrated)

File Name	Type	Primary Category	Top Process	Status	ID
No data found					

5 In the text field, enter the shared file location of the CSV SVN data input file to migrate.

6 Click the **Load** button.

7 Review the loaded data.

Note: After migration, flows with the status "Failed to Submit" are imported and placed in the "Unassigned" category. Users can cut and paste the flows to the proper categories in the Home page of the web editor application.

8 Choose a transformation option:

- **Truncate and Migrate:** To erase all data in the database, or if the database does not contain data.
- **Update and Migrate:** To retain data that is in the AnswerFlow table.

9 To migrate and transform objects from SVN to the AnswerFlow database table, click the **Migrate** button.

A window displays the message, “Data Migration in progress. Please wait...”

Wait for the migration process to finish. As a benchmark, a migration of 1400 objects takes about one hour.

When the migration process is complete, the imported data populates the migration table.

10 Click the **Export** button to put the migration log information into a spreadsheet for your review.

11 Review the migration log to look for any objects that failed the migration process (the status column indicates success or failure of each object's migration). Confirm whether top-level processes are being placed into the categories that you assigned. If necessary, confirm that the JSON is not corrupted.

12 After you fix any errors, run the migration again.

Note: Any processes not assigned to a category during the migration process are listed under the Unassigned link in the Object Library. A user who knows where the objects should be categorized can navigate to the Home page, select the unassigned objects, and then cut and paste these objects into the correct categories, apart from this migration procedure.

Starting and Stopping AnswerFlow

You can start and stop an AnswerFlow instance by using scripts that start and stop the application server for the instance.

Note: You must be within ICE (Integrated Collaboration Environment) to use the start and stop commands. *See Installing and Configuring Oracle Knowledge* for more information about ICE.

To start and stop the AnswerFlow instance:

1. Open a terminal window and navigate to `<answerflow_installation_directory>/instances/<instance_name>` where `<instance_name>` is the name of the instance, for example, Editor or Runtime.
2. Run the command to start ICE: `./setenv.sh`.
3. Start, stop, and restart the instance by executing the appropriate command:
   - `inquiraaf.sh start`
   - `inquiraaf.sh stop`
   - `inquiraaf.sh restart`

Enabling AnswerFlow to Work With Oracle Knowledge Analytics

As of Version 8.6, AnswerFlow is integrated with Oracle Knowledge Analytics (Analytics), an event-driven application that responds to system and user events. In AnswerFlow, an event captures the actions or occurrences that Analytics detects. Examples of event sources could include when a process call occurs, or when a process has reached a goal (goals being set manually by process flow editors as they configure individual process nodes).

Oracle Knowledge version 8.6 Analytics transforms and loads AnswerFlow event data to the reporting database. Analytics offers the following ready-to-use AnswerFlow reports:

- How many process flows have achieved their goals?
- Which process flows have been used most often?
- What is the rate at which process flows and goals have been achieved?
- Which steps in each process flow are being used most often?
- Which process flows have the highest abandonment rate?

To access these reports, sign into the Analytics module. For more information, see *Oracle Knowledge Analytics User's Guide*.

All Oracle Knowledge products send event messages by way of the Java Message Service (JMS) server queue, which is part of the Analytics application. Analytics logging is enabled for AnswerFlow upon product installation.

To enable Analytics logging, see “Modifying JMS Properties Files to Send Events to Analytics”. For more information, see *Oracle Knowledge Analytics Administrator's Guide*.

Verify by checking data in the `DW_STG_DATA` table under the `DW_STG` schema. The columns populated as part of this table are documented in *Oracle Knowledge Analytics User's Guide*.

Modifying JMS Properties Files to Send Events to Analytics

Configure AnswerFlow to send events to Analytics to modify the JMS properties to work with AnswerFlow:

1 Open the file `$ANSWERFLOW_INSTALLATION_DIRECTORY/Sample/SampleUI/src/main/resources/ok_jms.properties`

Tip: Optionally, you can copy the configured file `ok_jms.properties` from Information Manager in either the `lib` directory or the `conf` directory. This is useful if you have already set up Analytics for Information Manager and want to use the same configuration values.

2 Set the following properties as appropriate for the queue:

```
java.naming.security.principal=@JMS_USERID@

java.naming.security.credentials=@JMS_PASSWORD@

java.naming.provider.url=@JMS_URL@

ok.queue.connnection.factory.name=@JMS_FACTORY_NAME@

ok.queue.name=@JMS_QUEUE_NAME@
```

3 Set `ok.analytics.logging.enabled=`**`true`**.

4 Recompile and redeploy the SampleUI project by running the `deploy.sh` script from the Runtime ICE window.

Troubleshooting AnswerFlow

When difficult situations arise, you can refer to the error messages in AnswerFlow server log files to help diagnose problems. You can access these log files at the following locations:

```
<AnswerFlow installation directory>/instances/Editor(or RuntimeUI)/Inquira-Editor(or
RuntimeUI)-answerflow-<date>.log

<WebLogic domain directory>/servers/<server running AnswerFlow>/logs
```

The following table lists problem situations that might occur when using AnswerFlow, along with suggested solutions.

Issue	Solution
Some categories do not appear in AnswerFlow Editor.	Make sure that the category is configured for the channel and assigned to the user. For a category to be visible to a user, the category must be assigned to the AnswerFlow channel, and the category or a category above it hierarchically must be assigned to the user.
I cannot move a node where I want it in AnswerFlow Editor.	Uncheck the Snap to Grid box.
Some UI elements do not appear or are disabled for certain user roles.	As an AnswerFlow Administrator, verify the permissions for those roles in the AnswerFlow Configuration page.
Among multiple conditions from a branch, only one condition runs.	If multiple conditions outgoing from a branch are true, then the first condition in priority order runs. To make different branches run in different circumstances, compare at least one variable, rather than two constants, in the link conditions.

Issue	Solution
No nodes are visible when viewing or editing a process	Press and hold the Shift key while clicking the browser refresh button.
Effects on AnswerFlow of Applying an Oracle Knowledge upgrade	When an upgrade is applied, clear the cookies related to AnswerFlow and clear the browser cache.
I cannot find my promoted and published process in SampleUI.	The most recently promoted version of a promoted process should appear in the Latest tab of the SampleUI, unless the document containing the process has been reverted in Information Manager (in which case the version of the process contained in the version of the document that was reverted to would be displayed in the Latest tab). If the process does not appear in the Published tab, make sure that the process is published in Information Manager. Depending on the setup, you might have to go through a workflow to publish the process.
I see various issues when using one version of AnswerFlow Editor with a different version of AnswerFlow RuntimeUI.	Within a single environment, all instances of AnswerFlow Editor and RuntimeUI should be the same version. Data created in one version of AnswerFlow must be converted to the proper format for the new version according to the upgrade script before it is used in the new version (if the format has not changed, no conversion is necessary).
When using AnswerFlow Editor, an error page appears. The URL seems to contain unintelligible text at the end of it.	Delete everything in the URL after 'AnswerFlowEditor'. Press Enter to reload the page.
I see errors in RuntimeUI when running different versions of RuntimeUI that use the same memcached.	Each version of RuntimeUI should use its own instance of memcached.

CHAPTER 3

Composing Process Flows

Process designers (domain experts), business analysts, and developers can use this section to design and create process flows.

This section explains the tools and features on the Process page, where users view and edit processes. This page also contains links to preview and debug processes.

To reach the Process page:

1. Sign in to the AnswerFlow Editor web application.
2. From the Home page, you can go to the Process page in the following ways:
   a. Click the **Create** button, then choose Process or Subprocess to open the Process page in Edit mode with a new top-level process or a subprocess, respectively.
   b. Click the **View** or **Edit** link for a process in the table to open the Process page View or Edit mode, respectively, for that process.

Note: Some of the functionality described in the following sections about the Process page is also available on the Service Call page and the Context Variable page, and behaves similarly for those objects. This functionality includes: leaving the page, collapsing and restoring the left and right column panels, Cancel, Save, Save As, Submit, specifying object details, the Consumers list, Delete, Edit, Undo, and Redo.

How to Navigate the Canvas

From the Home page, you can open a Process object to open the canvas and see the Canvas features. The canvas navigation options work in either Edit mode or View mode. Opening this top-level process object, the canvas shows a Start node; a Home page with a Display Form, Display HTML, and Display IM Document; and three pages, each of which has page elements added with the page, or process step. The steps are linked based on how the user navigates through the flow activities.

The canvas is the area of the Process page where you place, move, edit and delete nodes. You can:

- Scroll through the entire flow by using the keyboard up, down, right, left, page up, page down, home, and end keys.
- Zoom in and zoom out of the canvas by using the mouse wheel. Note that the mouse wheel zooms in and out, it does not scroll.
- Navigate away from the window by using the module choices in the screen header (Home, Promote, Configuration, Data Migration) or by clicking Log Out at the upper right of the global screen header.

Leaving the Process Page

Navigate away from the Process page by using the module choices in the screen header (Home, Promote, Configuration, Data Migration) or by clicking Log Out at the upper right of the global screen header.

Note: Users only see the module navigation choices for features for which they are granted access. Most process designers, for example, do not see Configuration because they are not granted access to this functionality.

Actions in Edit Mode

The Process page in Edit mode provides a user with the following action options:

- Collapse and restore the Node Library panel and the Properties/Configuration/Variables panel using the small arrow control found at the inner edge of the panel.
- Cancel any edits made to the process without saving them.
- Save changes made to the process object.
- Save As, effectively making a copy of the process, which you will need to rename. Use this function to create copies of processes or to save processes as reusable subprocesses.
- Submit a completed process flow. Validation warnings may appear if there are any validation issues to correct.
- Specify process element details with right panel components to:
  - Specify process properties.
  - Specify the configuration details for the nodes and page elements on the canvas.
  - Name and configure local variables used in the process.
- Enable the Show Grid option to align the nodes in the canvas.
- Enable the Snap to Grid option to align nodes in the canvas along grid lines when they are positioned.
- Select objects on the canvas individually by clicking on a single object.
- Select multiple objects on the canvas by holding shift while clicking on them. You can select page elements only by selecting in a group with other elements on the same page.
- Move selected objects on the canvas by clicking and dragging.

Actions in View Mode

The Process page in View mode provides a user with the following options:

- The left navigation pane lists any consumed subprocesses, service calls, and context variables used in this process. It also lists whether any processes are consumers of the process currently being viewed. This section is always empty for top-level processes (which cannot be consumed by any other process). All items in the left navigation pane are active links that you can click and open in the current or a new browser tab.
- Actions in View Mode include:
  - Delete: You can delete processes that you have not submitted. After you submit a process successfully, you see an error message if you try to delete the object.
  - Edit: Switches to Edit mode for the process.
  - Export: Saves the canvas contents as a PDF of the complete canvas; allows you to review the process flow at a canvas level.The PDF does not contain any of the process properties or local variables, nor any of its node or page element configuration details. This action does not create an exportable process file for use in AnswerFlow.

  - Submit: Creates a set version of a process that can be referred to or reverted to. Submitted top-level processes can be compiled.
  - Debug: Allows you to walk through a preview of the process flow and observe the programmatic aspects of the process (variables passed, service calls used, context variable values). This is helpful for debugging flow problems.
  - Preview: Allows you to walk through a flow as it would be rendered for an end user. This is helpful for QA and UAT purposes.

Editing Tools

The editing tools available on the Process page include:

- Undo: Reverse or undo the last action taken on the Process page.
- Redo: Repeat the last action taken on the Process page.
- Delete one or more nodes or page elements: An item can be deleted by right-clicking on it and choosing delete from the menu or by selecting it on the canvas and then clicking the red X icon along the upper left of the canvas. If multiple items are selected, then the Delete option in the right-click menu and the red X icon along the upper left of the canvas will delete all of the selected items.
- Cut, copy, or paste one or more nodes or page elements: Selected items can be cut or copied by using the icons along the upper left of the canvas, by using the appropriate option in the right click menu, or by using a keyboard shortcut (Crtl-X or Cmd-X to cut, Ctrl-C or Cmd-C to copy). This places the selected items on the clipboard, which you can paste by using the Paste icon along the upper left of the canvas, by using the Paste option in the right click menu, or by using a keyboard shortcut (Crtl-V or Cmd-V).
- Link to or from a process element:
  - Right-click a node or page element, select Link From, then click a node or page element. A link from the first node or page containing the page element to the second node or page containing the page element is added, and any link previously coming from the node or page is removed.
  - Select a node or page element, click the Link From icon along the upper left of the canvas, then click a node or page element. A link from the first node or page containing the page element to the second node or page containing the page element is added, and any link previously coming from the node or page is removed.
  - Select a node or page element, right-click a different node or page element in a different node and select Link To. A link from the first node or page containing the page element to the second node or page containing the page element is added, and any link previously coming from the node or page is removed.
  - For links coming from a branch node: The previous methods apply, but instead of adding or moving the link immediately, a popup appears. You can then use the Link With button to decide which condition you want to link from the Branch node to the destination node with. This popup also allows you to add, delete, edit, and re-prioritize conditions.
  - Select a link, right click and select Change Link Destination or click the Change Link Destination icon along the upper left of the canvas, then click a node or page element. The link now points to the node or page containing the page element.
- Multi-select: Hold the shift key down while clicking on multiple items in the canvas to select them.
- Editing and navigation keyboard shortcuts:
  - Ctrl/Cmd-C: Copy
  - Ctrl/Cmd-X: Cut
  - Ctrl/Cmd-V: Paste

- Ctrl/Cmd-F: Search the Process (that is, the same as clicking the magnifying glass icon)
- Page Up: Pan the scene up one page (that is, the same height as the currently visible area)
- Page Down: Pan the scene down one page
- End: Pan the scene left one page
- Home: Pan the scene right one page
- Delete: Delete the currently selected nodes/display elements/links
- +/=: Zoom in
- -/_: Zoom out

Quick Search

Click the magnifying glass icon or press Ctrl/Cmd-F to open a quick search window in which you can search within a process. Enter the term you want to find. Limit which aspects of the process and which types of nodes you want to search for. You also have the options to match case and to search for whole words only.

Spelling Checker

Click the Check Spelling button to open the Spelling Checker tool. This is the same as the spelling checker in Information Manager.

A modal popup displays the terms noted as possible spelling errors. You can change the terms as needed, ignore them, replace them, and close the tool when done.

About Process Objects, Page Elements, and Logical Nodes

You create a process using process objects, pages, page elements, and logical nodes.

Process Objects

Process objects include processes, subprocesses, service calls, and context variables.

Processes

Processes are top-level processes that are compiled, promoted, published, and made available to end-user applications.

Subprocesses

Subprocesses are subsumed under top-level processes.

Service Calls

Service calls are Java calls that perform an action within, or retrieve data from, external resources, such as web services, file servers, and database servers, wherever data is stored that you want to bring into a service process. For example, you could use a service call to retrieve a customer's equipment model number from your customer database so that the process displays the most relevant solutions. This information is stored in a database that lists the equipment originally supplied to customers. A service call could connect to the database, retrieve a customer's information, and then display the information on-screen for the customer to confirm.

The concept of a service call is presented in three differing ways within AnswerFlow:

- Service Call Method: A Java method that a developer writes.
- Service Call Object: An AnswerFlow object that is stored in the AnswerFlow database. It contains the information about a Service Call Method so that the runtime knows how to carry out the method when a Service Call Node referencing that Service Call Object runs at runtime
- Service Call Node: A type of node, which references a Service Call Object.

A developer who creates service methods for use in AnswerFlow places the service JAR files in the classpath according to the type of server being used:

WEBLOGIC

- Editor: `$ANSWERFLOW_INSTALLATION_DIR/instances/Editor/webapps/AnswerFlowEditor/WEB-INF/lib`
- SampleUI: `$ANSWERFLOW_INSTALLATION_DIR/Sample/SampleUI/lib`

TOMCAT

- Editor: `$ANSWERFLOW_INSTALLATION_DIR/instances/Editor/appserveraf/webapps/AnswerFlowEditor/WEB-INF/lib`
- SampleUI: `$ANSWERFLOW_INSTALLATION_DIR/Sample/SampleUI/lib`

WEBSPHERE

- Editor & SampleUI: `<AnswerFlow installation directory>/AnswerFlowServicesLibrary`
- SampleUI only: `<AnswerFlow installation directory>/Sample/SampleUI/lib`

If a JAR in `<AnswerFlow installation directory>/AnswerFlowServicesLibrary` and a JAR in the currently deployed version of `<AnswerFlow installation directory>/Sample/SampleUI/lib` contain different versions of the same class, then:

- The class definition from the JAR in `<AnswerFlow installation directory>/AnswerFlowServicesLibrary` will be used for AnswerFlow Editor service calls.
- The class definition from the JAR in `<AnswerFlow installation directory>/Sample/SampleUI/lib` will be used for SampleUI service calls.

Note: For all server types, the Editor server must be restarted, and, for JAR files in `<AnswerFlow installation directory>/Sample/SampleUI/lib`, SampleUI must be redeployed before the new service methods will be usable in each application respectively. If the new service methods are not available after redeploying, you must restart the server. New methods in JAR files in `<AnswerFlow installation directory>/AnswerFlowServicesLibrary` will become available in each application when its WebSphere server is restarted.

CREATING A SERVICE CALL

To create a Service Call Object for use in the AnswerFlow Editor:

1 On the **Home** page, click the category in the **Object Library** list that you want to create the service call in.

2 Click the **Create** button and select **Service Call**.

The Creating New Service Call page opens.

3 In the **Name** field in the Properties pane, enter a unique, descriptive name to identify the service call.

4 Select the **Return Type**, either Bean or Map, to designate the type of information returned.

5 In the **Class Name** field, enter the Java class name.

6 In the **Method** field, enter the name of the Java Method used to retrieve the information, for example, `getCustomerInformation`.

7 If the method you are referencing has parameters:

a In the **Parameters** pane, click the **Add** button and enter the parameter Name.

b In the **Parameters** field, from the drop-down list select the parameter **Type**.

c Repeat step a and step b until you have set up all parameters to match the method being referenced.

Tip: In the Parameters pane Actions column, to delete a parameter, click X. To change the order of parameters, click the up or down arrow next to a parameter in the Actions column.

8 For return parameters settings, in the **Return Parameters** pane repeat step 7.

9 Click **Save**.

10 When you are ready to create a set version of this service call object that can be referred to or reverted to, click the **Submit** button.

Note: A process cannot be submitted until the service calls that it uses are submitted.

EDITING A SERVICE CALL

To edit a service call:

1 From the **Home** page, find the **Service Call** in the category you want to edit.

2 Click **Edit** in the action column for that object to open the Service Call screen.

3 Edit the fields as necessary.

4 Click **Save**.

5 When you are ready to create a set version of this service call object that can be referred to or reverted to, click the **Submit** button.

Note: A process cannot be submitted until the service calls it uses are submitted.

SERVICE CALL TECHNICAL DETAILS

When a Service Call node runs, the public static Java method (whose Class Name, Method, and Input Parameter Types align with the configurations in the Service Call Object) is loaded from the web application's classpath.

The Input Parameter values (strings in the AnswerFlow runtime data model) are converted before being passed to the method:

- If the parameter type is Object or String, no conversion occurs.
- If the parameter type is char, then the value is validated to be length 1, and the only char in the value is used.
- If the parameter type is numeric, then standard Java parsing with the wrapper class's static `valueOf` method (for example, `Integer.valueOf(String)`) is invoked.

Then the method runs with the converted Input Parameter values passed as arguments.

For each Output Parameter assigned to a variable in the Service Call Node, the value is retrieved from the return value of the method invocation.

If the Service Call Return Type is Bean, then the Output Parameter value is retrieved through reading the Java Bean property from the return value with the same Name as the Output Parameter. Assuming no custom Java Bean configuration, this will typically result in situations such as the int parameter named `exitCode` being retrieved by invoking the method named `getExitCode()` on the return value, and the boolean parameter named success being retrieved by invoking the method named `isSuccess()`.

Note: For the full JavaBeans specification, see `http://download.oracle.com/otndocs/jcp/7224-javabeans-1.01-fr-spec-oth-JSpec`.

If the Service Call Return Type is Map, then the Output Parameter value is retrieved through invoking the `Map.get(String)` method on the map, with the Output Parameter Name as an argument. The retrieved Output Parameter value is validated to be of the Type configured in the Service Call. Regardless of the Type, `toString()` is invoked before storing the result in the local variable configured to store the Output Parameter result.

If the execution produces errors (for example, the method does not exist on the classpath, or the input data is invalid for the required conversion), then the user sees a detailed error message in AnswerFlow Editor Preview and the ready-to-use SampleUI.

Context Variables

Create a Context Variable to bring data into a process based on its Variable Name. Context Variables can be populated through the following three methods (in decreasing precedence):

- Request parameter
- Request header
- Cookie

For example, if a customer is logged in and clicks a link to a process, the link might include a request parameter that indicates the customer's account number, which is then placed into a Context Variable that the process can access.

When creating a process, all defined Context Variables are stored and available to all users.

CREATING A CONTEXT VARIABLE

1. From the **Home** Page, click the **Create** button and select **Variable** from the drop-down list.
2. Name the variable, then provide the programmatic variable name.
3. Click **Save**, then click **Submit**.

EDITING A CONTEXT VARIABLE

1. From the **Home** page, find the Context Variable in the category you want to edit.
2. Click **Edit** in the actions column for that object.
   The Context Variable screen opens.
3. Make the changes needed to the Name or Variable Name fields.
4. Click **Save**.
5. When you are ready to create a set version of this Context Variable that can be referred to or reverted to, click the **Submit** button.

   Note: A process cannot be submitted until the Context Variables it uses are submitted.

Configuring Pages and Page Elements

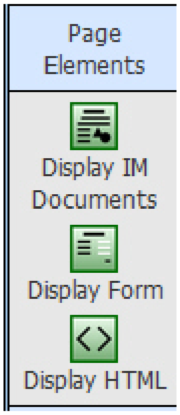
Process designers use pages as the container for composing specific, individual process steps in AnswerFlow. A page can contain one or more page elements.

About Pages

A Page is a process step container. One or more page elements are placed within a Page to compose a process step that could present knowledge articles, HTML content, and form controls for surveying the users or otherwise eliciting a decision.

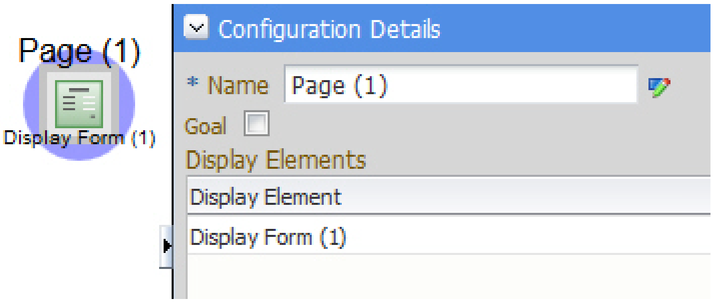
About Page Elements

Page elements are of three types: Display IM Documents, Display Form, and Display HTML.



Pages contain Page Elements (also called Display Elements). In most cases, you need to configure both the Page and its Page Elements. Although the following figures represent Pages containing Display Forms, the way of selecting elements also applies to Display HTML elements and Display IM Documents elements.

To select the Page, click the outside of the node icon. The Page is highlighted by a circle and the Configuration Details pane contains options for the Page.



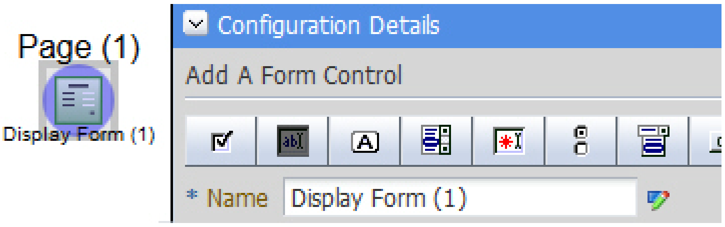
* Name Page (1)

Goal □

Display Elements

Display Element
Display Form (1)

To select the Display Element in a Page, select the center of the node icon. The center is highlighted by a circle and the Configuration Details pane contains options for the Page Element.



* Name Display Form (1)

Configuring Display Form Page Elements

You can configure forms to collect various types of information from users within a process using the Display Form page element. To display a form, add a Display Form page element to the process, select a form control from the Add A Form Control box, and then configure the form to request user input.

To add a Display Form page element:

1. If you are not adding the Display Form to an existing Page, select the node you want to link to the Page that will be created to contain the Display Form element.
2. Drag the **Display Form** page element onto a blank part of the canvas, or onto an existing Page if you want to add it to that Page.

   If you drop the Display Form onto a blank part of the canvas, AnswerFlow automatically creates a Page to contain it and (if the currently selected node has no outgoing link) adds a link from the currently selected node to the newly added Page.
3. In the **Name** field under Configuration Details, enter a unique name to identify the Display Form page element.
4. Enter the necessary parameters for each form control.
5. In the **Configuration Details** pane, shown in the previous figure, select **Form Controls** to display for the Page Element.
6. Configure the necessary parameters for each form control.

Configuring Form Control Parameters

You configure form control parameters when you add the form control to the process. You configure form control parameters after you add the form control to the process. Depending upon the form control added, you can configure different parameters. Not all parameters appear for every form control. The following table lists all of the parameters available for all of the form controls.

Parameter	Description
Label	Displays text about the form control.
Default Value	Populates the field with an optional default value when accessing the page.
Options	Provides user input options for various types of selections or menus, such as Check Box, Multi-Select, Radio Button, and Select. You indicate options by putting commas between them. For example, Option 1, Option 2, Option 3. The option that the user chooses at runtime is put into the Input Variable that is selected for this form control.
Input Variable	Corresponds to a variable in the Local Variables section of the page. AnswerFlow generates this variable automatically. You can select a different variable than the generated one. The data value that this form control receives goes into the selected Input Variable. **Tip:** Make the variable recognizable by renaming it as you define the form control. For example, if the field label is User Name, you might rename the variable as `username`.
Class Name	Identifies the CSS class to associate with the form control. This attribute is available for all form controls.
Required	Makes the element a required field in the user interface.
Tooltip	Describes the kind of data to enter if a field label needs additional description. For example, a page might have two fields, one labeled Customer Number and another labeled Account Number. A tooltip might be, “Five digit account number for individual customers.” for the first field and, “Eight digit account number for business customers.” for the second field.
Ghost Text	Appears as dimmed text in the field in the user interface. For example, for a Customer Name field you might designate how the entry should be made with the ghost text of “LAST,FIRST” in the field.
Image URL	Defines the URL of an image to use for the Submit button only.

Adding Display HTML Page Elements

AnswerFlow can display custom HTML at any point in a process. This is useful to display brief content that is not contained in an Information Manager document.

To configure a process to display custom HTML:

1 If you are not adding the Display HTML element to an existing Page, select the node you want to link to the Page that will be created to contain the Display HTML element.

2 Drag the **Display HTML** page element onto a blank part of the canvas, or onto an existing Page if you want to add it to that Page.

If you drop the Display HTML onto a blank part of the canvas, AnswerFlow automatically creates a Page to contain it and (if the currently selected node has no outgoing link) adds a link from the currently selected node to the newly added Page.

2 In the **Configuration Details** pane, enter a name for this page element.

3 Select **Constant** or **Variable** element.

4 If you selected Constant, in the text field, type the HTML code to use. If you selected Variable, then select a variable that will contain the HTML that you want to display.

Logical Nodes

A logical node is not visible to users at runtime. AnswerFlow provides the following types of logical nodes:

- Branch splits the process into divergent paths based on conditional logic, usually by evaluating a user response.
- Service Call uses a service call object to call a Java method, for external actions and retrieval of external data.
- Process Call references a previously defined process (subprocess), which then becomes part of the larger process.
- Set Variable Values sets the value of one or more variables.

Adding Branch Logical Nodes

Branch nodes divide a process based on one or more conditions, such as a user response, that the system evaluates.

To add a Branch node to a process:

1 Select the node that you want to link to the Branch node that you are going to add.

2 Drag the **Branch** node onto the canvas.

AnswerFlow automatically adds a link from the currently selected node to the newly added node (if the currently selected node has no attached link).

3 In the **Configuration Details** pane, enter a name for this node.

4 If Analytics is enabled, an AnswerFlowBranch event is logged any time a branch node is reached. To send an AnswerFlowGoalAchieved event to Analytics when the node is reached, check the **Goal** check box.

5 Select **Add** to create the first condition. You can add any number of conditions. Each condition that you add here corresponds to a link that is followed if the condition is true. If the condition is not true, the next condition is evaluated and, if true, the link corresponding to that condition is followed, and so on. If no conditions are true, an error occurs at runtime.

Select ✎ to open the Condition Builder and edit the conditions you have added.

Each row is a comparison of the left and right values that is evaluated at runtime as true or false. Each item may be either a constant typed in directly or a variable selected from the drop-down list whose value is used at runtime. In the middle drop-down list, the selected operator indicates how to compare the two values. To add more boxes, use the +OR button. To add more rows to a box, use the +AND button. At runtime, if all of the rows in a box are true, then the box evaluates to true. If any of the rows are false, the box evaluates to false. If any box is true, then the overall condition evaluates to true, and the corresponding link is followed, otherwise, the next condition is evaluated.

6 Select **Close** when completed.

Adding Service Call Logical Nodes

Use a service call when information from an external system, such as a database, is necessary or when you want to perform an action in an external system.

For more information, see “Creating a Service Call”.

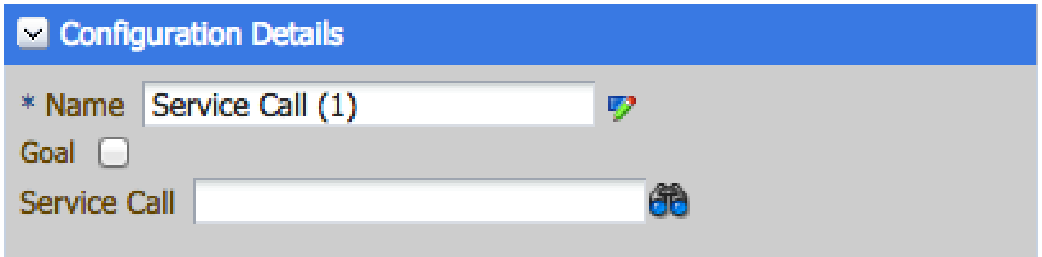
To add a service call node to a process:

1 Select the node you want to link to the Service Call node that you are about to add.

2 Drag the **Service Call** node onto the canvas.

AnswerFlow automatically adds a link from the currently selected node to the newly added node (if the currently selected node has no outgoing link).

3 In the **Name** field of the Configuration Details pane, enter a name for this node, as shown in the following figure.



Configuration Details

* Name Service Call (1)

Goal □

Service Call

4 If you know all or part of the name of the Service Call object you want to use in this Service Call node, enter the name in the text box next to **Service Call**, then click the **Search** icon. (The search is case-sensitive.)

5 In the popup, click **Select** next to the Service Call object that you want to use.

6 Pass data to the Subprocess using the Input Parameters and store data returned from the Subprocess by way of the Output Parameters in the process's local variables as needed.

Adding Set Variable Values Logical Nodes

To add a Set Variable Values node to a process:

1. Select the node you want to link to the Set Variable Values node that you are about to add.
2. Drag the **Set Variable Values** node onto the canvas.

   AnswerFlow automatically adds a link from the currently selected node to the newly added node (if the currently selected node has no outgoing link).
3. In the **Name** field of the Configuration Details area, type a name for this node.
4. Select **Add** to add a variable assignment to the node.

   At runtime, each assignment takes the value of the item in the Value column (either a constant or the value of the selected variable) and puts this value into the selected variable in the Variable column.

Adding Process Call Logical Nodes

Use the Process Call logical node to reuse a previously defined subprocess within a new, larger process.

To add a Process Call logical node:

1. Select the node you want to link to the Process Call node that you are about to add.
2. Drag the **Process Call** node onto the canvas.

   AnswerFlow automatically adds a link from the currently selected node to the newly added node (if the currently selected node has no outgoing link).
3. In the **Name** field of the Configuration Details area, type a name for this node.
4. If you know all or part of the name of the Subprocess you want to use in this Process Call node, enter it in the text box next to **Process**, then click the **Search** icon. (The search is case-sensitive.)
5. In the popup, click **Select** next to the Subprocess that you want to use.
6. Pass data to the Subprocess using the Input Parameters and store data returned from the Subprocess by way of the Output Parameters in the process's local variables as needed.

Previewing a Process

To preview the process:

1. Select a process or subprocess from the **Home** Page.
2. Click the **View** link in the **Action** column for that process.
3. On the Process page in View Mode, click the **Preview** link.

   The process is rendered as it would be in the SampleUI.
4. You can step through the process from your intended process user's point of view.

Debugging a Process

To debug a process:

1 Select a process or subprocess from the **Home** Page.

2 Click the **View** link in the **Action** column for that process object.

3 On the Process page in View Mode, click the **Debug** link.
As you move through the process, its logical steps are displayed in addition to the pages.

4 You can step through the process from both your intended process user's point of view, while looking at the programmatics aspects of the process flow that are usually invisible to the user.

Submitting a Process Object

You can submit process objects from their respective pages in Edit mode or View mode. Submit a process object to create a set version of it that can be reverted to and (in the case of Suprocesses and Service Calls) referred to. A process cannot be submitted until the process objects it uses are submitted. Submitted top-level processes can be compiled.

Deleting a Process Object

You can delete process objects before submitting them. After a process object is submitted, you cannot delete it. If you click the Delete button of a Submitted object, you get an error message noting you are not able to delete an object that is submitted.

Reverting a Process Object

If you need to revert to an earlier version of a process object:

1 Locate the object on the **Home** page.

2 Click the **Version** number link of the object.

3 In the Version modal popup, locate the row for the version to which you want to revert and click **View**.

4 In the object's View page, click the **Revert** button.
Clicking Revert creates a fresh version of the object, and it will be version X, where X is one number beyond the current version number. You can edit, save, and submit this version of the object.

Saving a PDF of the Canvas Window

On the Process page in View mode, when you click the Export button in the upper left of the screen, you generate a PDF snapshot of the canvas, at the Page/Page Element/Logical Node level. You can see the process's nodes, Page Elements, and how the process flows, but cannot see any node or element details other than name and type (by color/image).

CHAPTER 4

An Example of Creating a Process Flow

This section is an example of creating a process flow with various process objects and subprocesses. In this example, you are provided with instructions and details needed to build a complete top-level process flow from the ground up, using all AnswerFlow features.

Tip: For details on using the Editor canvas tools, see “Configuring Pages and Page Elements”.

Process Flow Best Practices

AnswerFlow is a visual modeling tool, similar to Microsoft Visio, which allows you to design your complete process flow. As a best practice, before you begin to plan your complete process flow, map out all process flow factors, including:

- All the decisions you must make and the choices for these decisions
- The process points where you will need to apply external data
- The process points where you could apply subprocesses
- The decision points where knowledge articles can answer questions and issues
- The decision points to escalate a question to an agent

This chapter walks you through each step of building a process, so you will not need to design a process.

Overview of the Process Flow Example

The example in this chapter leads you step-by-step to use all AnswerFlow functionality by building a process for a customer of a Heating and Air Conditioning service provider. The process allows the customer to sign up for a service contract on an HVAC (heating, ventilation, and air conditioning) unit. The process will help customers understand what level of service they can elect.

In this example, you will do the following activities:

- Create the Java service methods used by the process flow to access external data for decision-making.
- Create context variables needed in the process flow to confirm customer identity.
- Create the service calls used in two processes created in this example (a subprocess and top-level process).
- Create a reusable subprocess called Sign Up for Service.
- Create the top-level process called HVAC Service.

Create Service Methods

The first thing to do is create a Java class that will be packaged into a JAR file. These Java methods will make external system data available to the process flow that you will construct in this example.

The following code constitutes the class that contains the methods used in this example process flow.

Sample Java File

The following sample code is the content of the Java file used in this example process flow. Use the development environment of your choice to create the three Java methods illustrated in the following code example: `checkAccount, purchaseTime`, and `signUpForService`. Instructions and the location to upload the JAR file follow.

```
package com.inquira.pmeditor.servicecalls;

import java.text.SimpleDateFormat;
import java.util.Collections;
import java.util.Date;
import java.util.Map;

public class HvacService {

   public static AccountBean checkAccount(String username, String password)
   {
      return new AccountBean("Green", "Lennox Heating/Cooling Unit,"
      + "Desert Dry Dehumidifier, Cool Cuke Humidifier");

   }

   public static class AccountBean
   {
      private String accountStatus, registeredProducts;
      public AccountBean(String accountStatus, String registeredProducts) {
      this.accountStatus = accountStatus;
      this.registeredProducts = registeredProducts;
      }

      public String getAccountStatus() {
         return accountStatus;
      }

      public String getRegisteredProducts() {
         return registeredProducts;
      }
   }

   public static Map<String, Integer> purchaseTime(String month, String day,
   String year)
   {
      SimpleDateFormat sdf = new SimpleDateFormat("MMM d yyyy");
      try {
         Date date = sdf.parse(month + " " + day + " " + year);
         long timeMills = (new Date()).getTime() - date.getTime();
         long timeMillsInMonth = (long)(1000*60*60*24*(365242.199/1000/12));
         int months = (int)(timeMills/timeMillsInMonth);
         return Collections.singletonMap("months", months);
```

```
    } catch (Exception e) {e.printStackTrace();}
    return null;
}
public static Map<String, Boolean> signUpForService(String username, String
product, String level, String billingMethod, int daysToStartIn)
{
    return Collections.singletonMap("result", new Boolean(true));
}
}
```

Create Three Service Methods

Create three service methods and make them available to AnswerFlow:

1. Create a Java code file that contains the code shown in “Sample Java File”.
2. Compile the Java class into a JAR file.
3. To make the methods available to AnswerFlow Editor, place the JAR file as follows, depending on which server you are using:
   - WebLogic: `$ANSWERFLOW_INSTALLATION_DIR/instances/Editor/webapps/AnswerFlowEditor/WEB-INF/lib`
   - Tomcat: `$ANSWERFLOW_INSTALLATION_DIR/instances/Editor/appserveraf/webapps/AnswerFlowEditor/WEB-INF/lib`
   - WebSphere: `$ANSWERFLOW_INSTALLATION_DIR/AnswerFlowServicesLibrary`

   The service calls in the JAR become usable when you restart AnswerFlow Editor.
4. To make the methods available to AnswerFlow RuntimeUI:

   a. Place the JAR file as follows, depending on which server you are using:
   - WebLogic/Tomcat: `$ANSWERFLOW_INSTALLATION_DIR/Sample/SampleUI/lib`
   - WebSphere: `$ANSWERFLOW_INSTALLATION_DIR/AnswerFlowServicesLibrary`

   b. After you have placed the JAR file in the proper location, use the deploy script to rebuild the RuntimeUI application and redeploy it to the server.

Create Context Variables

Create two new context variables named *username* and *password*:

1. On the **Home** page, select a category.
2. Click the **Create** button and select **Variable**.
3. In the **Name** field, enter the text `username`.
4. In the **Variable Name** field, enter `username`. This will be used to retrieve the variable value from the context at runtime.
5. Click **Save**.
6. Click **Home** in the global header to return to the Home page.

7 Repeat step 2 through step 5 for the variable `password`.

Create Three Service Calls

Create three service calls named *checkAccount*, *purchaseTime*, and *signUpForService*.

Create Service Call *checkAccount*

1 On the **Home** page, click the **Create** button and select **Service Call** to open the Creating New Service Call screen.

2 In the **Class Name** field, enter the full class name, which is the package name of the class with the class name. Because the example Java class in the JAR file has the package name `com.inquira.pmeditor.servicecalls` and the class name is `HvacService`, enter `com.inquira.pmeditor.servicecalls.HvacService`. All three methods will use this same class name.

3 In the **Name** field in the right side **Properties** pane, enter the method name. In this example it is ***checkAccount***.

4 For **Return Type**, select **Bean**.

5 Enter the parameter names that are specified in the JAR file:

a Under the **Parameters** pane, click the **Add** button to add a row to enter a parameter.

b In the **Name** field, enter ***checkAccount*** and select **String** from the Type drop-down list.

c Click the **Add** button to add another row and then enter ***purchaseTime*** in the **Name** field and select **String** from the Type drop-down list.

d Click the **Add** button to add another row. Enter ***signUpForService*** in the **Name** field, and then select **String** from the Type drop-down list.

6 Enter the return parameter names that are specified in the Java file:

a Under the **Return Parameters** pane, click the **Add** button to add a row to enter a return parameter.

b In the **Name** field, enter the return parameter ***accountStatus*** and select **String** from the **Type** drop-down list.

c Click the **Add** button to add another row.

d In the **Name** field, enter the return parameter ***registeredProducts*** and select **String** from the **Type** drop-down list.

7 Click the **Save** button.

Create Service Call *purchaseTime*

1 Add the service call method *purchaseTime* by repeating step 1 and step 2 in “Create Service Call checkAccount”.

2 In the **Name** field in the right side **Properties** pane, enter ***monthsAgo***.

This name will be visible in AnswerFlow. The method name in the Java file is *purchaseTime*.

3 Enter the parameter names that are specified in the Java file.

a ***Month:*** String

b ***Day:*** String

c ***Year:*** String

4 Enter the return parameter names that are specified in the Java file:

a Under the **Return Parameters** pane, click the **Add** button to add a row to enter a return parameter.

b In the **Name** field, enter the return parameter ***months*** and select **Int** from the **Type** drop-down list.

c For **Return Type**, select **Bean**.

5 Click the **Save** button.

Create Service Call *signUpForService*

1 Add the service call method *signUpForService* repeating step 1 and step 2 in “Create Service Call checkAccount”.

2 In the **Name** field in the right side **Properties** pane, enter ***signUpForService.***

3 For Return Type, select **Map**.

4 Enter the parameter names that are specified in the JAR file:

a ***username***: String

b ***product***: String

c ***level***: String

d ***billingMethod***: String

e ***daysToStartIn***: Int

5 In the Return Parameters pane, enter the return parameter name ***result*** of Type **boolean**.

6 Click the **Save** button.

Create the Subprocess *Sign Up for Service*

A subprocess is a process that enters from and returns to another process. A subprocess cannot be a top-level process. Any top-level process can reuse a subprocess.

You will create the subprocess according to the following procedures:

- Set a variable value.
- Create a page with form elements and HTML elements to get the customers sign-up details.
- Add a branch and specify conditions to either fire a service call or set a variable.

You will add this subprocess to the top-level process HVAC Service, which you will build later.

Create a Subprocess

To create the subprocess:

1 On the **Home** page, click the **Create** button and select **Subprocess** to open the Editor canvas window.

2 In the **Properties** pane on the right side, in the **Name** field enter the name ***Sign Up for Service*.**

3 In the **Node Library** on the left side, drag the **Set Variable Values** icon and drop it near the **Start** node.

The node remains selected and its details populate the Configuration Details pane on the right side.

4 In the **Name** field of the **Configuration Details** pane, enter ***decline=false*** to indicate that this node will set the variable *decline* to *false*.

Set a Variable

1 In the **Node Library** on the left side, drag the **Set Variable Values** icon and drop it near the **Start** node.

2 Under the **Local Variables** pane, click the **Add** button to add a row.

3 In the **Name** field, enter the variable name ***decline***.

4 Under the **Configuration Details** pane, click the **Add** button to add a variable row.

5 In the **Configuration Details** pane, select **decline** from the **Variable** drop-down list.

6 Leave the **Value** type at the default **Constant** and enter ***false*** in the text field so that the variable *decline* equals *false*.

Create the *Sign Up for Service* Page

Create a page that has three HTML page elements and two form control page elements.

1 Drag a Page to the Editor canvas.

2 Add the User display element to the page:

Tip: Dropping a page element onto the Editor canvas creates a page to contain it. Dropping page elements on top of an existing page adds those page elements to the page.

a Drag and drop a **Display HTML** element onto the page.

The element remains selected and its details populate the Configuration Details pane on the right side.

b In the **Configuration Details** pane **Name** field, enter ***User***.

c Click **Variable.**

d From the drop-down list, select the variable ***username****.*

3 Add the Product display element:

a Drag and drop a **Display HTML** element onto the page.

b In the **Configuration Details** pane **Name** field, enter ***Product***.

c Under the **Local Variables** pane, click the **Add** button to add a row.

d In the **Name** field in the added row, enter ***product*** and click **Is Input?** to indicate that the value for this variable will be passed into this subprocess when it is called.

e In the **Configuration Details** pane, from the drop-down list select the variable **product**.

4 Add the Level of Service display element:

a Drag and drop another **Display HTML** display element.

b In the **Configuration Details** pane **Name** field, enter ***Level of Service,*** and then click **Variable**.

c Under the **Local Variables** pane, click the **Add** button to add a row.

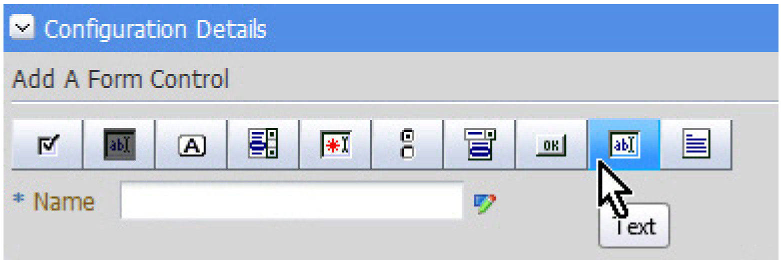
d In the **Name** field in the added row, enter **level** and click **Is Input?** to indicate that the value for this variable will be passed into this subprocess when it is called.

e In the **Configuration Details** pane, from the drop-down list select the variable **level**.

5 Add the Enter Info element.

a Drag and drop a **Display Form** display element onto the page. Name this element ***Enter Info****.*

b In the **Configuration Details** pane, click the **Text** button, as shown in the following figure.



c In the **Configuration Details** pane, scroll down to the **Text** section.

d In the **Label** field, enter ***Days from now to start service (0 for today)***.

e Under the **Local Variables** pane, click the **Add** button to add a row.

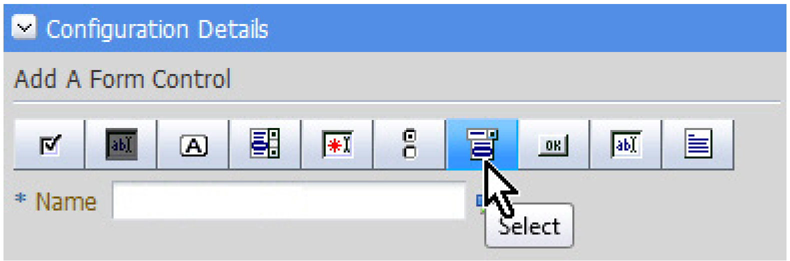
f In the **Name** field in the added row, enter ***daysToStartIn***.

g Check the **Is Input?** check box to indicate that the value for this variable will be passed into this subprocess when it is called.

h In the **Configuration Details** pane in the **Text** section, from the Input Variable the drop-down list select the variable **daysToStartIn**.

i Click the **Select** button, as shown in the following figure. This will add a Select form control to the Display Form.

A section to configure it appears.



j In the **Configuration Details** pane, scroll down to the **Select** section.

k In the **Label** field, enter ***Billing Preference***.

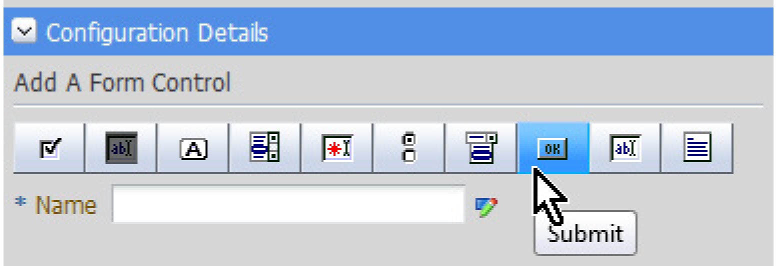
l In the **Options** field, enter ***Charge me monthly, Charge me quarterly, Charge me annually***.

m In the **Local Variables** pane, add ***BillingMethodSelected***.

6 Add the “I wish to decline service” element:

a Drag and drop a **Display Form** display element onto the page. Name this node ***I wish to decline service***.

b In the **Configuration Details** pane, click the **Submit** button, as shown in the following figure.



c Scroll down to the **Submit** section. In the **Label** field, enter ***Decline***.

d In the **Input Variable** field, select **decline** from the drop-down list.

7 Select the page (click the title or the outside borders of the node) and rename the page ***Sign Up For Service***.

Add a Branch Node

Add a branch node to the subprocess and then set up two conditions. In the first condition, the variable *decline* is compared with *equals(string)* to the constant *Decline*. For the second condition, two constant *1*s are compared for equality.

To add a branch node:

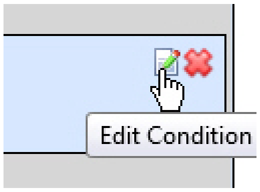
1 Select the **Sign Up for Service** page.

2 Drag and drop a **Branch** node onto the Editor canvas.

3 In the **Configuration Details** pane, in the **Name** field enter ***Decline Service?***

4 Add two conditions to the branch node by clicking the **Add** button twice under **Conditions** in the **Configuration Details** pane.

5 Under **Conditions**, click the **Edit Condition** icon of the first condition.



The Condition Builder dialog box opens.

6 Do the following:

- From the left drop-down list, select the variable **decline**.
- From the center drop-down list, select **equals(string)**.
- In the right text field, enter ***Decline***.

7 Click the **Close** button.

8 Click the **Edit Condition** icon of the second condition.

- Click the left radio button labeled **Constant**.
- Enter ***1*** in the left text field.
- Select = in the center drop-down list.
- Enter ***1*** in the right text field.

Set a Variable

1 Select the **Branch** node.

2 Drag and drop a **Set Variable Values** node onto the Editor canvas.

3 In the **Configuration Details** pane, in the **Name** field enter ***result=false***.

4 In the **Local Variables** section, click **Add** to add a row.

5 In the new row, enter ***result*** in the **Name** field and check the **Is Output?** check box.

6 Under the Configuration Details pane, click the **Add** button to add a variable row.

7 In the Configuration Details pane, select **result** from the Variable drop-down list.

8 Leave the Value type at the default **Constant** and enter ***false*** in the text field so that the variable result equals false.

Add a Service Call

1 Select the **Branch** node again.

a Drag and drop a **Service Call** node onto the Editor canvas.

b In the **Configuration Details** pane, enter ***Process Service Sign Up*** in the **Name** field.

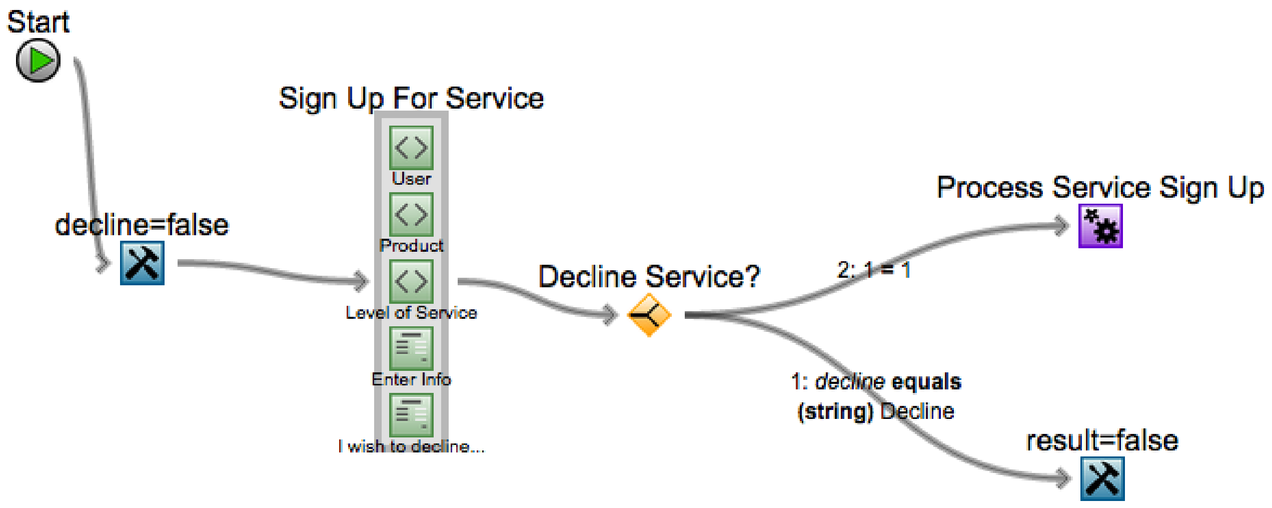
c In the **Input Parameters** section, click **Variable** for each parameter.

d From the **Value** drop-down lists, for each parameter select the value that corresponds to the name of the parameter, for example, for *product* select ***product***, and for *billingMethod* select ***billingMethodSelected***.

2 Click the **Save** button.

The Sign Up For Service process flow should resemble the following figure.

Validate Your Subprocess



In the prior section, you saved your subprocess and the canvas window switched from "Edit" to "View" mode. You are now able to preview and debug this process.

1 Click the **Debug** button.

2 Step through your subprocess to test it.

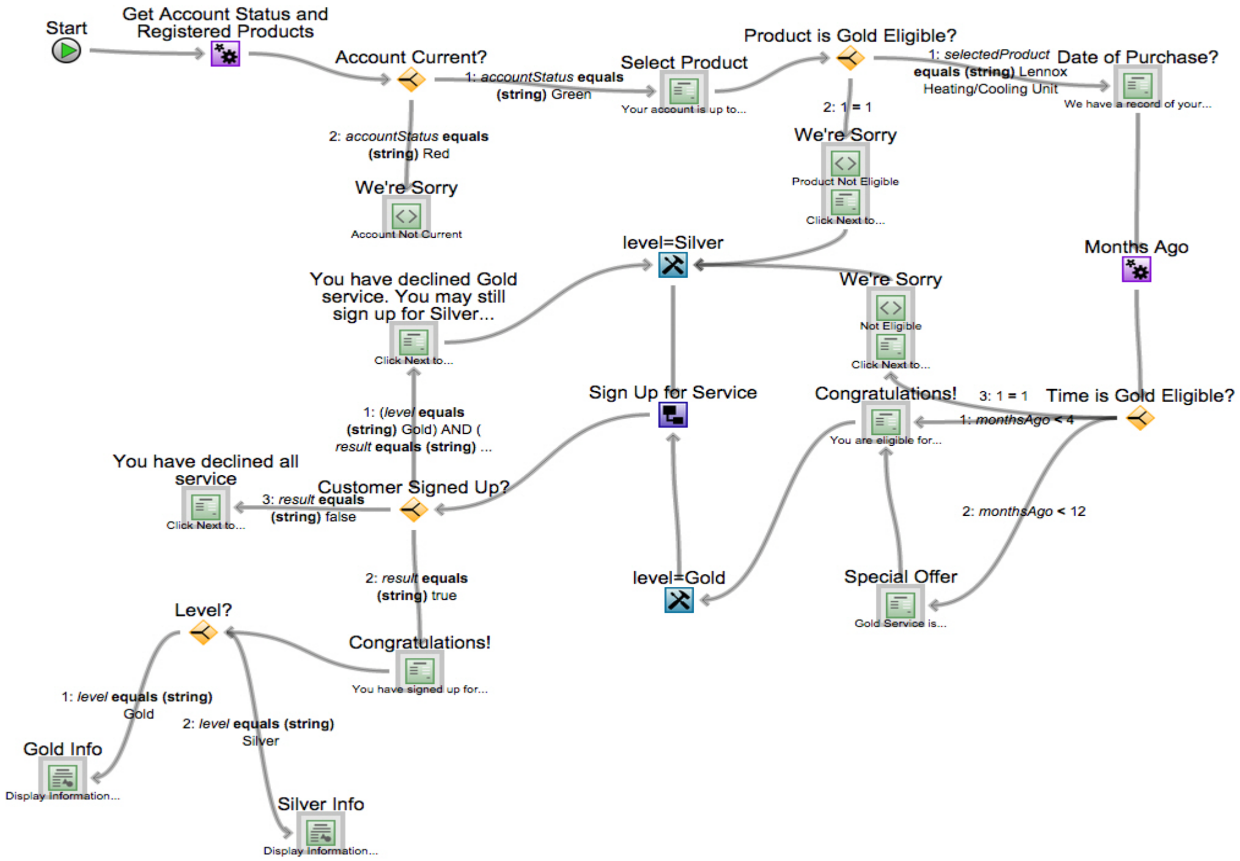
3 Fix any errors. When you are satisfied that the subprocess is correct, click **Submit**.

The subprocess is ready for use in your top-level process.

For more information, see “Debugging a Process”.

Create the Top-Level Process *HVAC Service*

Using the components you have already created, you will create a top-level process. The completed process flow should resemble the following diagram.


Start
Get Account Status and Registered Products
Account Current?
1: accountStatus equals (string) Green
Select Product
Your account is up to...
Product is Gold Eligible?
1: selectedProduct equals (string) Lennox Heating/Cooling Unit
Date of Purchase?
We have a record of your...
2: accountStatus equals (string) Red
We're Sorry
Account Not Current
2: 1 = 1
We're Sorry
Product Not Eligible
Click Next to...
level=Silver
Months Ago
You have declined Gold service. You may still sign up for Silver...
Click Next to...
We're Sorry
Not Eligible
Click Next to...
Sign Up for Service
Congratulations!
You are eligible for...
3: 1 = 1
Time is Gold Eligible?
1: monthsAgo < 4
1: (level equals (string) Gold) AND (result equals (string) ...
You have declined all service
Click Next to...
Customer Signed Up?
3: result equals (string) false
2: monthsAgo < 12
2: result equals (string) true
level=Gold
Special Offer
Gold Service is...
Level?
Congratulations!
You have signed up for...
1: level equals (string) Gold
2: level equals (string) Silver
Gold Info
Display Information...
Silver Info
Display Information...

Create the Process

1. Click the **Home** button.
2. Click the **Create** button and select **Process**.
3. In the **Properties** pane, name the process ***HVAC Service.***

Start the Process with a Service Call

1. Drag and drop a **Service Call** node onto the Editor canvas.
2. In the **Name** field of the **Configuration Details** pane, enter ***Get Account Status and Registered Products***.
3. In the **Service Call** field of the **Configuration Details** pane, click the search icon to open the Select a Service Call list.
4. Select **checkAccount**.
5. The parameters panes and Version field appear.
6. In the **Input Parameters** pane, for the parameter ***username***, click **Variable.**

7 Open the Context Variables drop-down list and select **username**.

8 In the **Input Parameters** pane, for the parameter ***password*** click **Variable** to open the Context Variables drop-down list and select **password**.

9 In the **Local Variables** pane, click the **Add** button to add a row.

10 In the **Name** field, enter ***accountStatus***.

11 Add another row and in the **Name** field enter ***registeredProducts***.

12 In the **Output Parameters** pane, in the drop-down lists select **accountStatus** and **registeredProducts** for the corresponding parameters.

Confirm the Account is Current

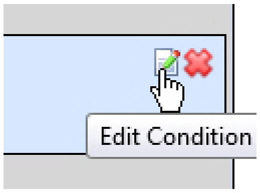
1 Drag and drop a **Branch** node.

2 In the **Name** field of the Configuration Details pane, enter ***Account Current?***

3 With the Branch node selected, drag and drop a **Display Form** display element and a **Display HTML** display element.

In the Configuration Details pane, under the Conditions section, two conditions with Priority 1 and Priority 2 appear.

4 Click the **Edit Condition** icon of the first condition to open the Condition Builder dialog box.



5 In the **Condition Builder** dialog box:

- From the left drop-down list, select the variable **accountStatus**.
- From the center drop-down list, select **equals (string)**.
- In the right text field, enter ***Green***.

6 Click **Close**.

7 Click the **Edit Condition** icon for the second condition to open the Condition Builder dialog box.

- From the left drop-down list, select the variable **accountStatus**.
- From the center drop-down list, select **equals (string)**.
- In the right text field, enter ***Red***.

8 Click **Close**.

9 Select the page containing the Display HTML and rename it ***We're Sorry.***

10 Select the Display HTML in the page and rename it ***Account Not Current***.

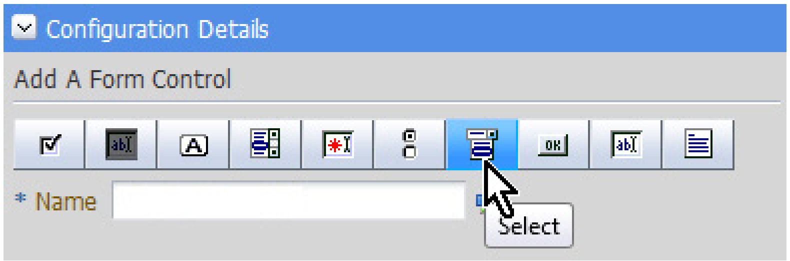
11 In the text area under Configuration Details, enter ***Your account is not current; a payment is due. Please bring your account up to date and then you can try this process again.***

Confirm the Product is Gold Eligible

1 Select the page containing the Display Form and name it ***Select Product.***

2 Select the **Display Form** in the Page. In the **Configuration Details** pane, in the **Name** field, enter ***Your account is up to date***. Select the product for which you want to add Gold Service.

3 On the tool bar ribbon, click the **Select** button.



A Select form control is added is added to the Display Form, and a section to configure it appears.

4 In the **Local Variables** pane, click the **Add** button to add a row, and then name it ***selectedProduct***.

5 In the **Select** form control section, **General Properties** tab, enter the following:

- **Options**: Click **Variable** and select ***registeredProduct***.
- **Input Variable**: Select **selectedProduct**.

6 With the Select Product page or Display Form inside it selected, drag and drop a branch node onto the canvas. In the **Configuration Details** pane rename the branch node to ***Product is Gold Eligible?***

7 Under **Conditions**, click the **Add** button twice to add two conditions:

a For the first condition, click the edit icon to open the Condition Builder dialog box:

- From the left drop-down list, select the variable **selectedProduct**.
- From the center drop-down list, select **equals (string)**.
- In the right text field, enter ***Lennox Heating/Cooling Unit***.

b Click the **Close** button.

c For the second condition, click the edit icon to open the Condition Builder dialog box:

- Above the left drop-down, select the **Constant** radio button. Then enter ***1*** in the text field.
- From the center drop-down list, select **=**.
- In the right text field, enter ***1***.

d Click the **Close** button.

Confirm Purchase is Gold Eligible

1 With the **Product is Gold Eligible** branch node selected, drag and drop a **Display Form** next to it. Name the newly added page ***Date of Purchase?***

2 Select the Product is Gold Eligible branch node, then drag and drop a **Display HTML** element next to it. Name the newly added page ***We're Sorry***.

- **a** Click the **Display HTML** element in the We're Sorry page and name it ***Product Not Eligible.***
- **b** In the text area in the **Configuration Details** pane, enter ***This product is not eligible for Gold Service. You may sign up for Silver Service instead.***

3 Drag and drop a new **Display Form** and add it to the **We're Sorry** page by releasing it while the cursor is within the boundaries of the page. Then name it ***Click Next to Continue***.

- **a** Select the display form element in the **Date of Purchase?** page. In the **Name** field, enter ***We have a record of your purchase, but we don't have your product registration. Please enter your purchase date.***
- **b** Click the **Select** button on the tool bar ribbon three times.
- **c** In the first **Select** form control configuration section, enter ***Month*** in the **Label** field.
- **d** In the **Options** field, enter the names of all twelve months, each month separated by a comma.
- **e** In the **Local Variables** pane:
  - Click the **Add** button and enter the Name ***monthSelected*** in the added row.
  - Click the **Add** button and enter the Name ***daySelected*** in the added row.
  - Click the **Add** button and enter the Name ***yearSelected*** in the added row.
- **f** In the **Input Variable** field, select **monthSelected** from the drop-down list.
- **g** In the second **Select** form control configuration section, enter ***Day*** in the **Label** field.
- **h** In the **Options** field, enter the dates of a month, 1 through 31, with each number separated by a comma.
- **i** In the **Input Variable** field, select **daySelected** from the drop-down list.
- **j** In the third **Select** form control configuration section, enter ***Year*** in the **Label** field.
- **k** In the **Options** field, enter the years 2015 through 2010, with each year separated by a comma.
- **l** In the **Input Variable** field, select **yearSelected** from the drop-down list.

4 With the **Date of Purchase?** page selected, drag a **Service Call** node and drop it near the page.

- **a** In the **Configuration Details** pane, enter ***Months Ago*** in the **Name** field.
- **b** Click the **Search** icon, then click **Select** next to the **monthsAgo** service call.
- **c** In the **Input Parameters** section, for the parameters **Month**, **Day**, and **Year**, select the corresponding variable, for example, **monthSelected** for Month.
- **d** In the **Local Variables** pane, add a variable named ***monthsAgo***.
- **e** In the **Configuration Details** pane, in the **Output Parameters** table, select the variable **monthsAgo** as the variable for the output parameter months.

5 With the **Months Ago** service call node selected, drag a **Branch** node in and drop it near the Months Ago service call node. Name it ***Time is Gold Eligible?***

6 Add three conditions:

- Priority 1: monthsAgo < 4, where *monthsAgo* is a variable and 4 is a constant.
- Priority 2: monthsAgo < 12, where *monthsAgo* is a variable and 12 is a constant.
- Priority 3: 1 = 1, where 1 is a constant.

7 With the branch node selected, drag and drop two **Form Control** nodes and one **Display HTML** element.

8 Specify the following details:

a Select the page pointed to by the link labeled **2: monthsAgo < 12** and name it **Special Offer**.

b Select the **Display Form** element in the page and name it ***Gold Service is typically offered only for products purchased in the last four months; but right now we have a special offer. Click Next to accept.***

c Right-click on the display element of the other **Form Control** node. From the drop-down list, select **Link To** to connect it to the **Special Offer** form control node.

d Select the page pointed to by the link labeled **1: monthsAgo < 4**. Name it ***Congratulations!***

e Select the display form element in this page and name it ***You are eligible for Gold Service!***

9 With the **Congratulations!** page selected, drag and drop a **Set Variable Values** node and name it ***level = Gold***.

a In the **Configuration Details** pane, scroll to the bottom and click **Add**.

b In the added row, select **level** as the **Variable** and enter ***Gold*** as the **Value**.

10 With the **level = Gold** node selected, drag and drop a **Process Call** node and name it ***Sign Up For Service***.

a In the **Process** field, select the search icon to open the Select a Process dialog box.

b Select **Sign Up for Service**.

c In the **Input Parameters** section, select **Variable** for each parameter's value. For the parameter `level`, select **level**. For the parameter `product`, select **selectedProduct**.

d In the **Local Variables** pane, add the variable ***result***.

e Select **result** in the **Output Parameters** section.

Create the *Sign Up for Service* Conditions and Dialog

1 With the **Sign Up For Service** node selected, drag and drop a **Branch** node and name it ***Customer Signed Up?***

2 Add three conditions:

a For the first condition, click the **Edit Condition** button to open the Condition Builder.

- From the left drop-down list, select the variable **level**.
- From the center drop-down list, select **equals (string)**.
- In the right text field, enter ***Gold***.

b Click the **+AND** button below the green row. In the added green row:

- From the left drop-down list, select the variable **result**.
- From the center drop-down list, select **equals (string)**.
- In the right text field, enter ***false***.

3 With the branch node selected, drag and drop three **Display Form** elements.

a Select the first of the three pages that were added. Enter the name ***You have declined Gold service. You may still sign up for silver service.***

b Select the display form element of the first page and name it ***Click Next to Continue.***

4 With the display element still selected, drag and drop a **Set Variable Values** node and name it ***level = Silver***.

5 In the **Variable Assignments** section, scroll to the bottom and click the **Add** button. In the added row, select **level** as the variable and enter ***Silver*** as the value.

6 Right click the **level=Silver** Set Variable Values node, select **Link From**, and then click the **Sign Up for Service** Process Call node.

7 Go back to the two **We're Sorry** pages, which contain both a Display HTML and a Display Form element (but not the one that contains only a Display HTML element) and link them to the **level=Silver** Set Variable Values node.

Tip: To link nodes, right-click a node, select **Link From**, then click the node that you want to link to.

Confirm Customer is Signed Up

1 Select the page linked from the branch node **Customer Signed Up?** with the link labeled **3: result equals (string) true**.

a Name the page ***You have declined all service***.

b Select the Display Form element in the page and enter the name ***Click Next to Continue***.

2 Select the remaining page node linked to the branch node Customer Signed Up?

a Name the page ***Congratulations!***

b Select the Display Form element in the page and name it ***You have signed up for service!***

Provide Service Knowledge By Level

1 Select the **Congratulations!** page and drag and drop a **Branch** node. Name the Branch node ***Level?***.

2 Add two conditions:

- Priority 1: **level equals (string) Gold**, where *level* is a variable and *Gold* is a constant.
- Priority 2: **level equals (string) Silver**, where *level* is a variable and *Silver* is a constant.

3 Select the new branch node and drag and drop two **Display IM Documents** elements.

a Select the page linked from the branch node **Level?** with the link labeled **1: level equals (string) Gold**. Name the page ***Gold Info***.

b Click the **Display IM Documents** element in the page and name it **Display Gold Info Document**.

c In the **Type a Doc ID** field, enter the ID of an Information Manager document and click **Add**.

Note: For the purposes of this example, do not create an imitation or invalid Information Manager document, as it will become part of your searchable Knowledge base.

4 Name the remaining page ***Silver Info***. Name the element in the page ***Display Silver Info Document***.

a Select the page linked from the branch node **Level?** with the link labeled **2: level equals (string) Silver**. Name the page ***Silver Info***.

b Click the **Display IM Documents** element in the page and name it **Display Silver Info Document**.

c In the **Type a Doc ID** field, enter the ID of an Information Manager document and click **Add**.

Save, Preview, and Debug the Process

1 Click **Save** to create the process flow.

2 Debug the process and preview it. For more information, see “Debugging a Process” and “Previewing a Process”.

3 Promote the process. See “Promoting Processes”.

CHAPTER 5

Deploying Processes

You can deploy AnswerFlow process flows by:

- Compiling a submitted, validated process flow from the AnswerFlow Home page
- Promoting a successfully compiled process flow from the Promote page
- Publishing a promoted process flow from the Information Manager console

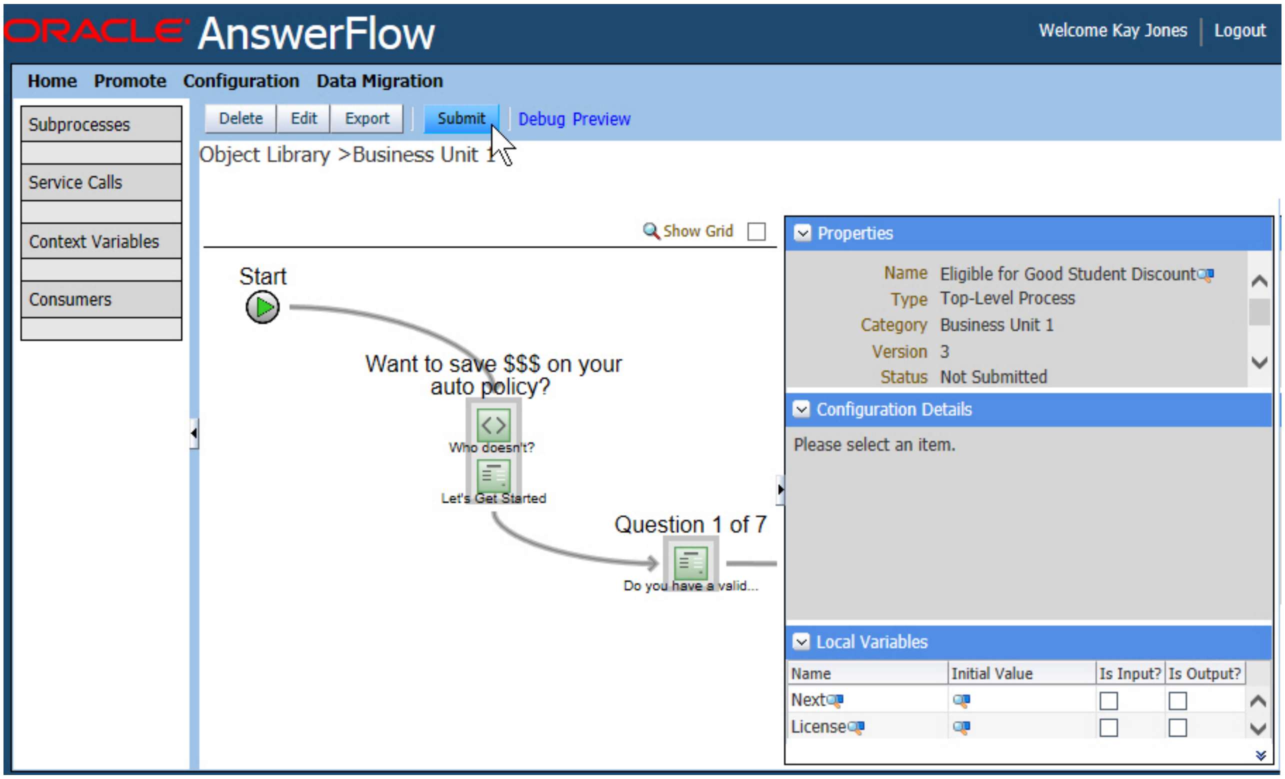
This chapter presents the steps to deploy processes, along with details of how to unpublish processes.

Compiling Processes

To compile a process flow, you must first submit the process from the Process page. Submitting a process creates a set version of the process that can be referred to or reverted to. You can compile submitted top-level processes.

To submit processes:

1. On the Home page, click the **Edit** button or the **View** button to open the process flow in the Canvas window.
2. Click the **Submit** button to open a Submission Notes window.



Name Eligible for Good Student Discount
Type Top-Level Process
Category Business Unit 1
Version 3
Status Not Submitted

Please select an item.

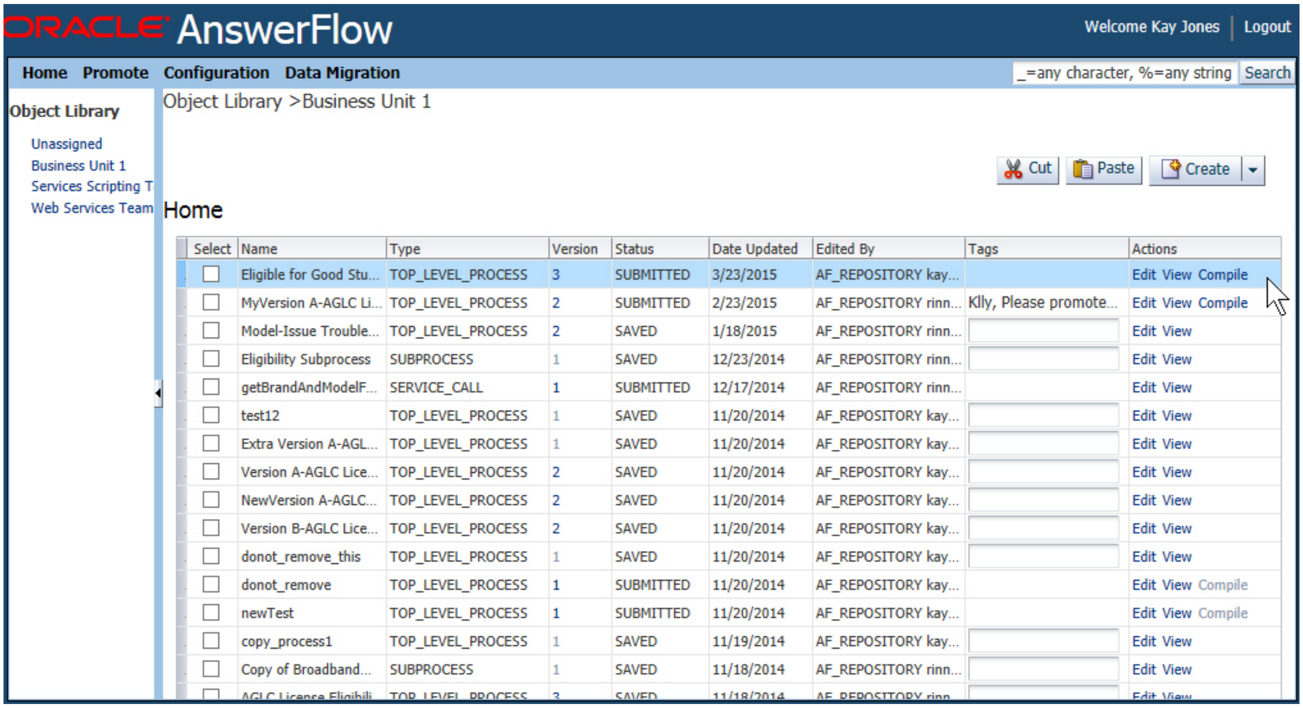
Name	Initial Value	Is Input?	Is Output?
Next		□	□
License		□	□

3 In the Submission Notes window, add a required note and click **OK**.

A confirmation message appears at the top of the screen.

When you navigate to the Home page, the object's status is Submitted and a Compile link is in the Actions column of the object's table entry.

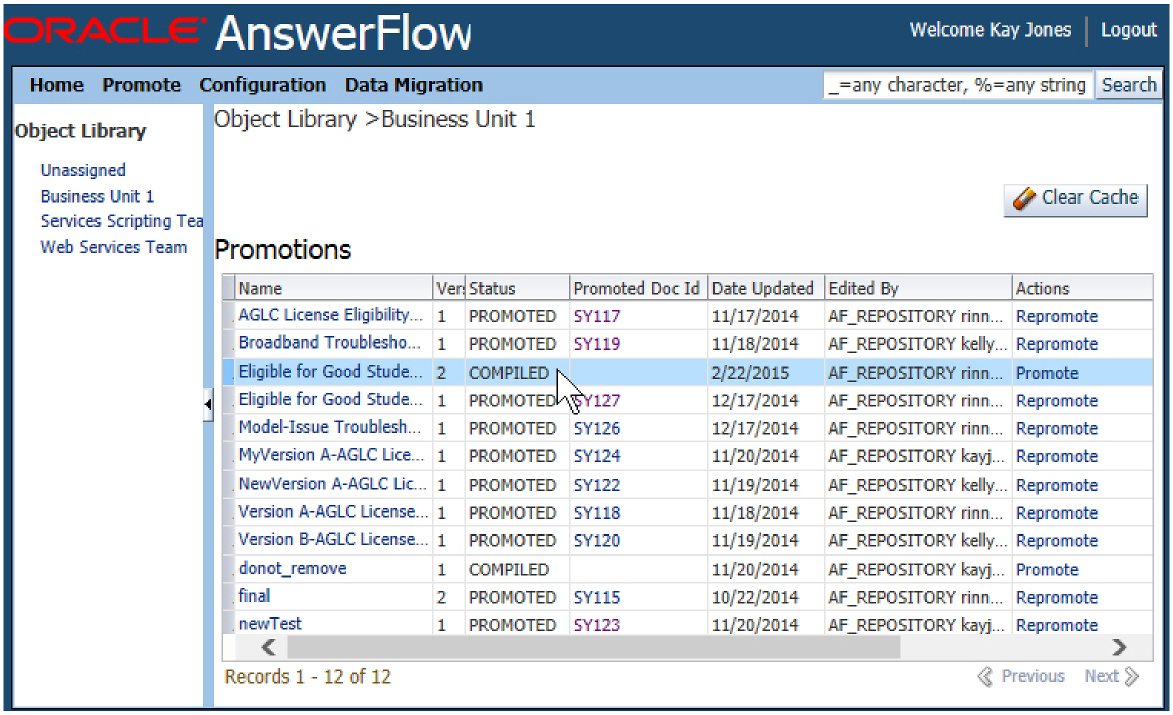
After you submit a top-level process from the Process page, it has a Compile link in the Actions column on the Home page, as shown in the following figure.


ORACLE AnswerFlow
Welcome Kay Jones | Logout
Home Promote Configuration Data Migration
_=any character, %=any string
Search
Object Library
Unassigned
Business Unit 1
Services Scripting T
Web Services Team
Object Library >Business Unit 1
Cut
Paste
Create
Home

Select	Name	Type	Version	Status	Date Updated	Edited By	Tags	Actions
□	Eligible for Good Stu...	TOP_LEVEL_PROCESS	3	SUBMITTED	3/23/2015	AF_REPOSITORY kay...		Edit View Compile
□	MyVersion A-AGLC Li...	TOP_LEVEL_PROCESS	2	SUBMITTED	2/23/2015	AF_REPOSITORY rinn...	Klly, Please promote...	Edit View Compile
□	Model-Issue Trouble...	TOP_LEVEL_PROCESS	2	SAVED	1/18/2015	AF_REPOSITORY rinn...		Edit View
□	Eligibility Subprocess	SUBPROCESS	1	SAVED	12/23/2014	AF_REPOSITORY rinn...		Edit View
□	getBrandAndModelF...	SERVICE_CALL	1	SUBMITTED	12/17/2014	AF_REPOSITORY rinn...		Edit View
□	test12	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Extra Version A-AGL...	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Version A-AGLC Lice...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	NewVersion A-AGLC...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	Version B-AGLC Lice...	TOP_LEVEL_PROCESS	2	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	donot_remove_this	TOP_LEVEL_PROCESS	1	SAVED	11/20/2014	AF_REPOSITORY kay...		Edit View
□	donot_remove	TOP_LEVEL_PROCESS	1	SUBMITTED	11/20/2014	AF_REPOSITORY kay...		Edit View Compile
□	newTest	TOP_LEVEL_PROCESS	1	SUBMITTED	11/20/2014	AF_REPOSITORY kay...		Edit View Compile
□	copy_process1	TOP_LEVEL_PROCESS	1	SAVED	11/19/2014	AF_REPOSITORY kay...		Edit View
□	Copy of Broadband...	SUBPROCESS	1	SAVED	11/18/2014	AF_REPOSITORY rinn...		Edit View
□	AGLC License Eligibili...	TOP_LEVEL_PROCESS	3	SAVED	11/18/2014	AF_REPOSITORY rinn...		Edit View

4 To compile a top-level process, find it on the Home page and click its active **Compile** link.

The Promotions page appears, containing the process with status Compiled, as shown in the following figure.


ORACLE AnswerFlow
Welcome Kay Jones | Logout
Home Promote Configuration Data Migration
_=any character, %=any string
Search
Object Library
Unassigned
Business Unit 1
Services Scripting Tea
Web Services Team
Object Library >Business Unit 1
Clear Cache
Promotions

Name	Ver	Status	Promoted Doc Id	Date Updated	Edited By	Actions
AGLC License Eligibility...	1	PROMOTED	SY117	11/17/2014	AF_REPOSITORY rinn...	Repromote
Broadband Troublesho...	1	PROMOTED	SY119	11/18/2014	AF_REPOSITORY kelly...	Repromote
Eligible for Good Stude...	2	COMPILED		2/22/2015	AF_REPOSITORY rinn...	Promote
Eligible for Good Stude...	1	PROMOTED	SY127	12/17/2014	AF_REPOSITORY rinn...	Repromote
Model-Issue Troublesh...	1	PROMOTED	SY126	12/17/2014	AF_REPOSITORY rinn...	Repromote
MyVersion A-AGLC Lice...	1	PROMOTED	SY124	11/20/2014	AF_REPOSITORY kayj...	Repromote
NewVersion A-AGLC Lic...	1	PROMOTED	SY122	11/19/2014	AF_REPOSITORY kelly...	Repromote
Version A-AGLC License...	1	PROMOTED	SY118	11/18/2014	AF_REPOSITORY rinn...	Repromote
Version B-AGLC License...	1	PROMOTED	SY120	11/19/2014	AF_REPOSITORY kelly...	Repromote
donot_remove	1	COMPILED		11/20/2014	AF_REPOSITORY kayj...	Promote
final	2	PROMOTED	SY115	10/22/2014	AF_REPOSITORY rinn...	Repromote
newTest	1	PROMOTED	SY123	11/20/2014	AF_REPOSITORY kayj...	Repromote

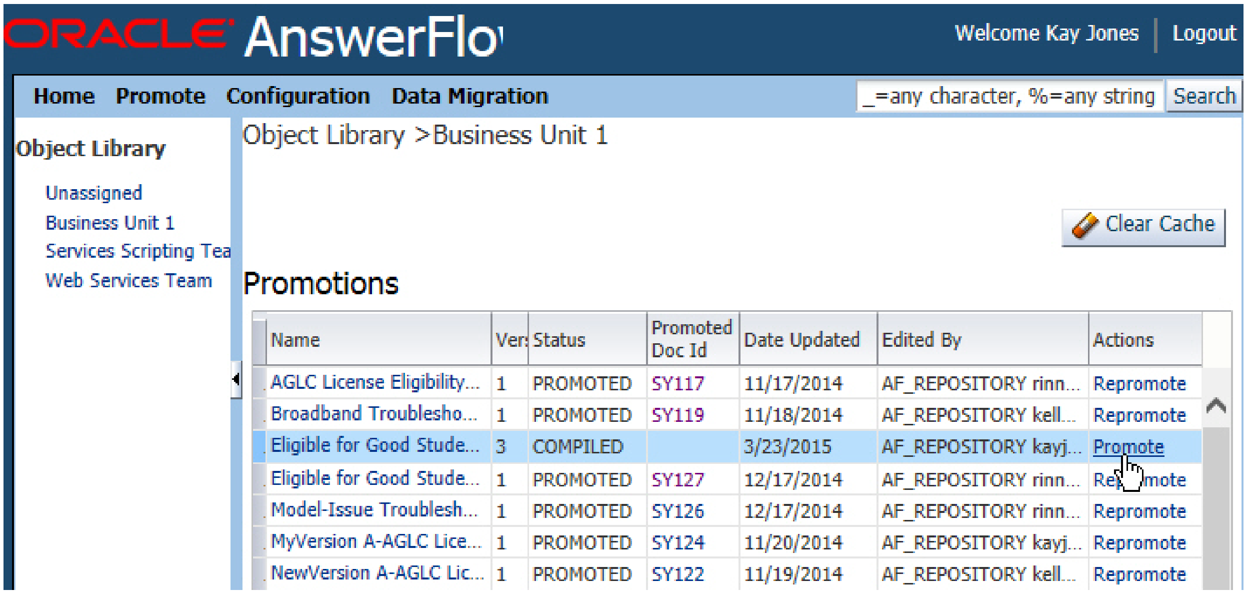
Records 1 - 12 of 12

Promoting Processes

From the Promote page, you can promote processes with the status Compiled so that you can later publish them in Information Manager. You can promote only processes with the status Compiled.

To promote a process:

1 In the Promotions page, click the **Promote** link of the process.


ORACLE AnswerFlo
Welcome Kay Jones | Logout
Home Promote Configuration Data Migration
_=any character, %=any string
Search
Object Library
Unassigned
Business Unit 1
Services Scripting Tea
Web Services Team
Object Library >Business Unit 1
Clear Cache

Promotions

Name	Ver	Status	Promoted Doc Id	Date Updated	Edited By	Actions
AGLC License Eligibility...	1	PROMOTED	SY117	11/17/2014	AF_REPOSITORY rinn...	Repromote
Broadband Troublesho...	1	PROMOTED	SY119	11/18/2014	AF_REPOSITORY kell...	Repromote
Eligible for Good Stude...	3	COMPILED		3/23/2015	AF_REPOSITORY kayj...	Promote
Eligible for Good Stude...	1	PROMOTED	SY127	12/17/2014	AF_REPOSITORY rinn...	Repromote
Model-Issue Troublesh...	1	PROMOTED	SY126	12/17/2014	AF_REPOSITORY rinn...	Repromote
MyVersion A-AGLC Lice...	1	PROMOTED	SY124	11/20/2014	AF_REPOSITORY kayj...	Repromote
NewVersion A-AGLC Lic...	1	PROMOTED	SY122	11/19/2014	AF_REPOSITORY kell...	Repromote

The status of the process changes from Compiled to Promoted and a numbered link appears in the Promoted Doc ID column. The promotion process wraps an AnswerFlow process in an Information Manager document so that it can be executed in the AnswerFlow Runtime.

Re-Promoting Processes

If you promote multiple versions of a process, and then decide to revert it to a previous version, you can re-promote the previous version of the process.

Process flows can be re-promoted when you want the latest version of the document containing your process flow in Information Manager to have its System Data Node contain the AnswerFlow-specific data from a specific version that has already been promoted, but you do not want to change the attributes of the Information Manager document outside the System Data node.

If it is not important to preserve the attributes outside of the System Data node, you can roll back the Information Manager document to the version associated with an older process, instead of using the Repromote feature.

To re-promote a process:

In the Promotions page, click the **Repromote** link in the Actions column of the process.

The Repromote link dims.

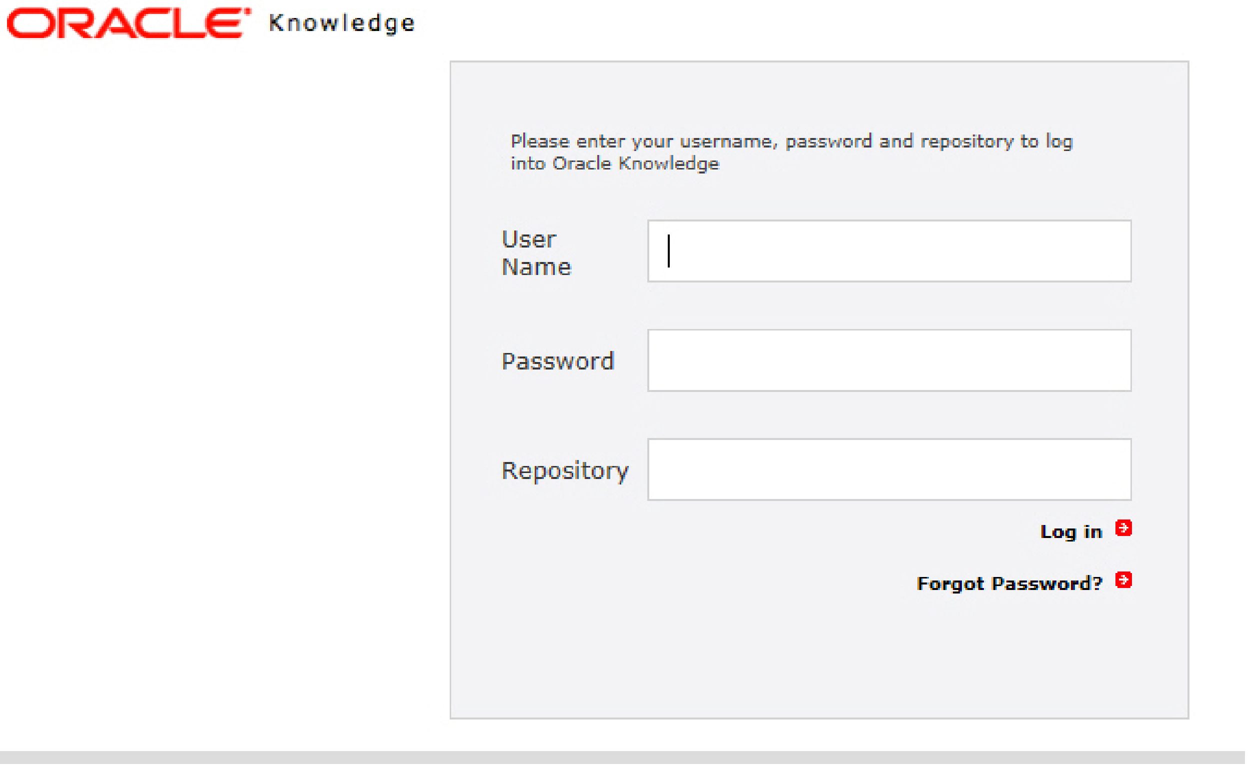
Publishing Processes Within Information Manager

You publish process flows within Information Manager.

To work with a process flow in Information Manager:

1 Navigate to the AnswerFlow **Promote** page and find the promoted process you want to publish.

2 Click the **DocID** link of the process to open the Information Manager login page.

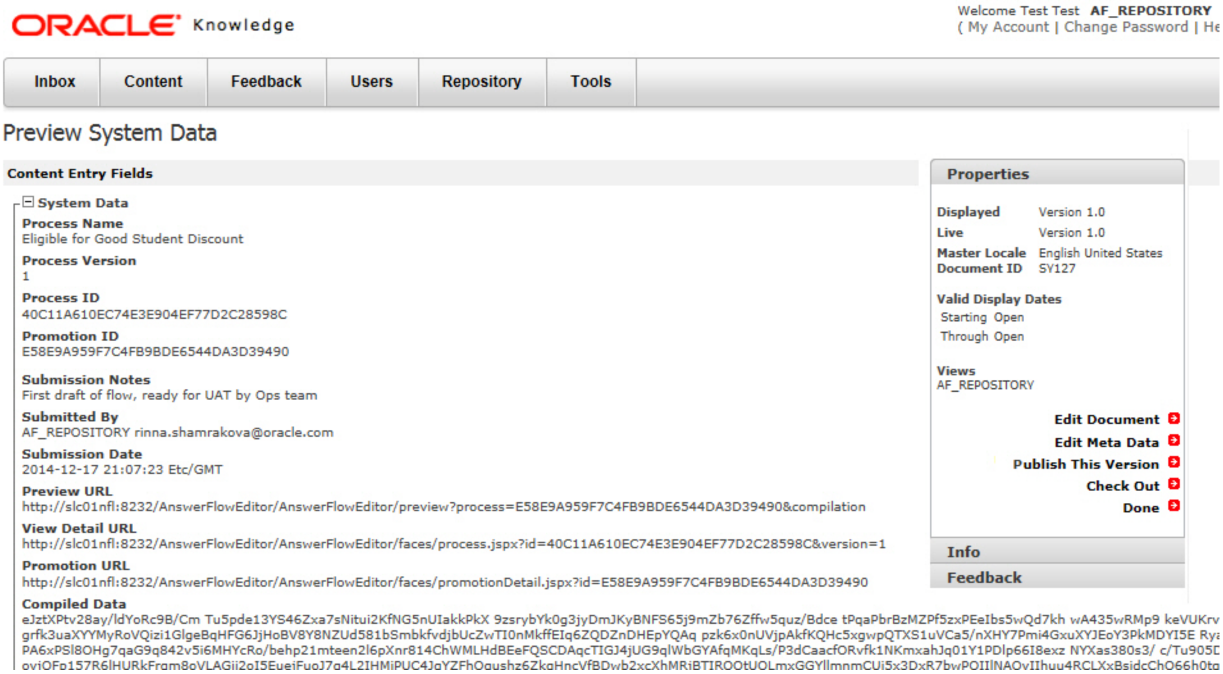


Please enter your username, password and repository to log into Oracle Knowledge

3 Enter the same username, password, and repository that you used to log in to AnswerFlow. Confirm that the selection in the Repository field is correct.

4 Click **Log in** to open the process's Information Manager AnswerFlow Channel document screen.

The system data node is expanded so you can see how the process data maps to the node attributes. This screen is not responsive, so you may just see the data, as illustrated below, and would have to scroll to see the publishing actions on the right side of the screen.


ORACLE Knowledge
Welcome Test Test AF_REPOSITORY
(My Account | Change Password | He
Inbox
Content
Feedback
Users
Repository
Tools

Preview System Data

Content Entry Fields

⊟ System Data

Process Name
Eligible for Good Student Discount

Process Version
1

Process ID
40C11A610EC74E3E904EF77D2C28598C

Promotion ID
E58E9A959F7C4FB9BDE6544DA3D39490

Submission Notes
First draft of flow, ready for UAT by Ops team

Submitted By
AF_REPOSITORY rinna.shamrakova@oracle.com

Submission Date
2014-12-17 21:07:23 Etc/GMT

Preview URL
http://slc01nfl:8232/AnswerFlowEditor/AnswerFlowEditor/preview?process=E58E9A959F7C4FB9BDE6544DA3D39490&compilation

View Detail URL
http://slc01nfl:8232/AnswerFlowEditor/AnswerFlowEditor/faces/process.jspx?id=40C11A610EC74E3E904EF77D2C28598C&version=1

Promotion URL
http://slc01nfl:8232/AnswerFlowEditor/AnswerFlowEditor/faces/promotionDetail.jspx?id=E58E9A959F7C4FB9BDE6544DA3D39490

Compiled Data

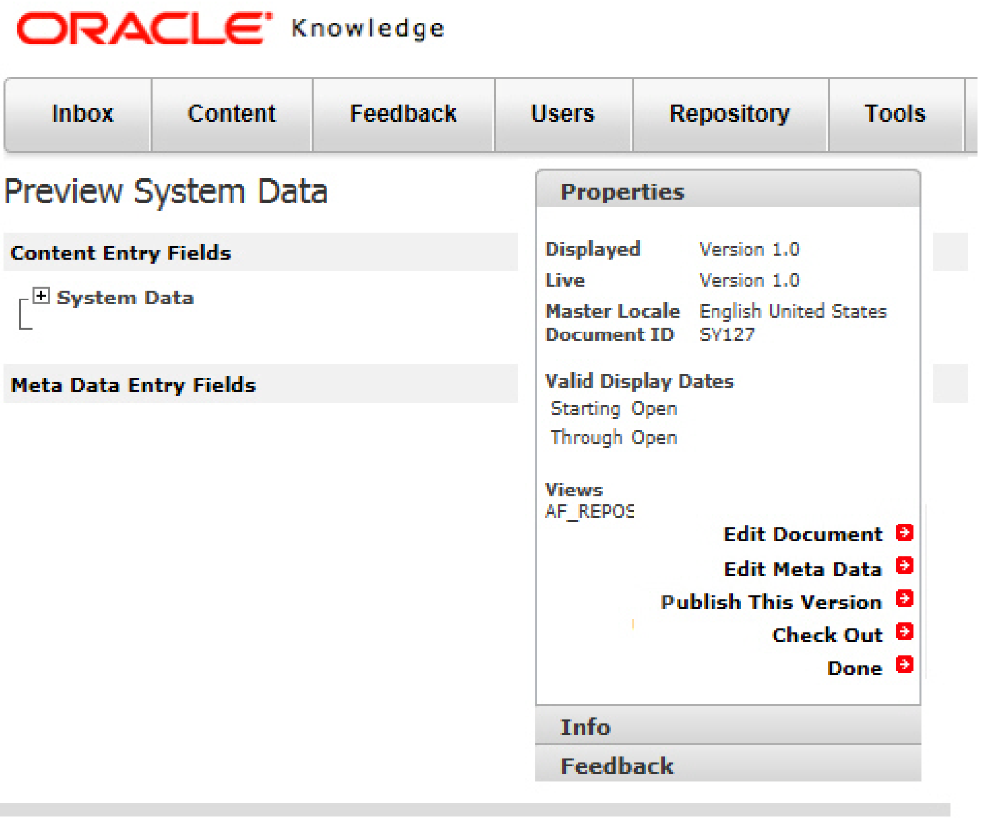
eJztXPtv28ay/ldYoRc9B/Cm Tu5pde13YS46Zxa7sNitui2KfNG5nUIakkPkX 9zsrybYk0g3jyDmJKyBNFS65j9mZb76Zffw5quz/Bdce tPqaPbrBzMZPf5zxPEeIbs5wQd7kh wA435wRMp9 keVUKrv
grfk3uaXYYMyRoVQizi1GlgeBqHFG6JjHoBV8Y8NZUd581bSmbkfvdjbUcZwTI0nMkffEIq6ZQDZnDHEpYQAq pzk6x0nUVjpAkfKQHc5xgwpQTXS1uVCa5/nXHY7Pmi4GxuXYJEoY3PkMDYI5E Ry
PA6xPSl8OHg7qaG9q842v5i6MHYcRo/behp21mteen2l6pXnr814ChWMLHdBEeFQSCDAqcTIGJ4jUG9qlWbGYAfqMKqLs/P3dCaacfORvfk1NKmxahJq01Y1PDlp66I8exz NYXas380s3/ c/Tu905D
oviOFp157R6lHURkFrqm8oVLAGii2oI5EueiFuoJ7q4L2IHMiPUC4JqYZFhOqushz6ZkqHncVfBDwb2xcXhMRiBTIROOtUOLmxGGYllmnmCUi5x3DxR7bwPOIlNAOvIIhuu4RCLXxBsidcChO66h0tq

Displayed	Version 1.0
Live	Version 1.0
Master Locale	English United States
Document ID	SY127

Valid Display Dates
Starting Open
Through Open

Views
AF_REPOSITORY

Tip: You can collapse the node to remove unwanted data from view.



Preview System Data

Content Entry Fields

⊞ System Data

Meta Data Entry Fields

Displayed	Version 1.0
Live	Version 1.0
Master Locale	English United States
Document ID	SY127

Valid Display Dates
Starting Open
Through Open

Views
AF_REPOS

You may see other nodes (such as document title) or other parts of the channel form if they have been defined for your AnswerFlow Channel. The figure illustrated above only has the required System Data node. Also, if your AnswerFlow Channel is designed to use a workflow, you would see the workflow links instead of a Publish This Version link.

5. To publish this process flow (assuming you do not have a workflow set up for the AnswerFlow Channel), click the **Publish This Version** link.

   The process appears under the Published tab in the SampleUI dashboard.

6. If you have Memcached installed and you have published another version of this process in the past, return to the AnswerFlow application. If you did not log out or close the browser tab containing AnswerFlow, you can click that tab to return to AnswerFlow.
7. Go to the **Promote** page.
8. Click the **Clear Cache** button to refresh the SampleUI cache.

Unpublishing Processes Within Information Manager

To remove a process flow from production access:

1. Sign in to AnswerFlow.
2. Navigate to the **Promote** page.
3. Locate the process you want to unpublish and click its **Doc ID** link.
4. Sign in to Information Manager to open the Preview page for the document that contains the process.
5. Click the **Unpublish This Version** link.

   The process is no longer available for viewing in SampleUI. The Information Manager Properties panel now indicates that there is no Live process available.

6. Exit Information Manager.
7. If you have Memcached installed, return to the AnswerFlow application. If you did not log out or close the browser tab containing AnswerFlow, you can click that tab to return to AnswerFlow.
8. Navigate to the **Promote** page.
9. Click the **Clear Cache** button to refresh the SampleUI cache.

CHAPTER 6

Integrating Processes in End-User Interfaces

When you promote AnswerFlow processes, they are available to the AnswerFlow RuntimeUI. You can view them by using the SampleUI Dashboard, a ready-to-use web application that displays all published process flows you have created. The SampleUI Dashboard also provides access to the latest versions of promoted process flows, enabling you to perform user acceptance testing before publishing.

The audiences for this chapter are developers and other IT technical staff who are responsible for integrating AnswerFlow process flows into production web applications. In this section we provide instructions to help you understand how to compile and deploy SampleUI, so that you can either customize its interface or integrate it into production web applications to deliver process flows to your end-users.

This section describes how to deliver process flows to your end users in two main ways: SampleUI and Search.

Using SampleUI to Deliver Process Flows to End-Users

You can deliver process flows to end-user interfaces by using SampleUI to:

- Embed SampleUI into Information Center or another production web application.
- Customize SampleUI source code to create a production web application.
- Use SampleUI as a JSON web service.

Compiling and Deploying SampleUI

Before using SampleUI for the first time, you must compile and deploy it to the RuntimeUI server.

Location of SampleUI project assets (JAR files and source code):

```
$ANSWERFLOW_INSTALLATION_DIR/Sample/SampleUI/lib
```

Depending on your operating system, run the following command in a RuntimeUI ICE window:

- Windows: Run `deploy.bat`
- Other operating system: Run `deploy.sh`

Note: The first time you run this script, you must specify the source directory and WAR file, using the -s and -w parameters. For example, the first time you run `deploy.sh` for an unmodified SampleUI, the following command would work:

```
deploy.sh -s $INSTALL_DIR/Sample/SampleUI -w $INSTALL_DIR/Sample/SampleUI/target/
SampleUI.war
```

Embedding the SampleUI Dashboard in Web Applications

To embed the SampleUI Dashboard into Information Center or other web applications, follow the instructions in this topic.

Requirements

- Get the file flow-8.5.1.2.3.zip from My Oracle Support.
- Locate the Information Center web application files in your instance.
- Set up and configure the .jsp pages within flow-8.5.1.2.3.zip to use SampleUI.
- Embed process flows within InfoCenter pages.
- Troubleshoot as needed.

Locate the Information Center Web application Files

Locate the Information Center web application files; $INFOCENTER_WEBAPP is at the following locations, per the application server used in your Oracle Knowledge instance.

- WebLogic
  `$INFOMANAGER_INSTALLATION_DIRECTORY/instances/$INFOMANAGER_INSTANCE_NAME/webapps/$INFOCENTER_WEBAPP_NAME/app`
- Tomcat
  `$INFOMANAGER_INSTALLATION_DIRECTORY/instances/$INFOMANAGER_INSTANCE_NAME/appserverim/webapps/$INFOCENTER_WEBAPP_NAME`
- WebSphere
  `$WEBSPHERE_APPSERVER_INSTALLATION_DIRECTORY/profiles/$INFOMANAGER_PROFILE_NAME/installedApps/$INFOMANAGER_CELL_NAME/infocenter.ear/infocenter.war`

Note: You can integrate SampleUI into a web application other than InfoCenter by using a similar method to the one described in step 2 through step 4 of the following section. If the host system supports embedding templated or raw HTML, the integrator can convert `c_flow.jsp` into a supported template or raw HTML file and integrate the converted file into the host system. The rest of the non-InfoCenter-specific steps are the same.

Setting Up and Configuring SampleUI

1. Unzip `flow-8.5.1.2.3.zip` as `$INFOCENTER_WEBAPP/apps/infocenter/custom/pages/flow`.
2. Edit `$INFOCENTER_WEBAPP/apps/infocenter/custom/pages/flow/c_flow.jsp` to fill in the SampleUI host and port in the assignment to `sampleUiUrl` at the top of the file.
3. Navigate to the Configuration page in AnswerFlow Editor and set `sampleUi.domainAndPort` to the protocol, domain, and port of SampleUI (for example, `http://example.com:8233`) as it would be accessed through a web browser (for example, not a domain name that is only resolvable from the Information Manager server.)

Note: After you set this property, do not try to directly access SampleUI through a browser at any other domain and port, or problems may occur. For instance, if SampleUI is accessible at both `http://server-name.domain-name:8233/SampleUI` and `http://server-name:8233/SampleUI`, and you set `config.domainAndPort=http://server-name:8233`, then you should access SampleUI only through `http://server-name:8233/SampleUI`, and not through `http://server-name.domain-name:8233/SampleUI`.

4 Restart the Information Manager and AnswerFlow RuntimeUI servers.

Embedding Process Flows

To render AnswerFlow processes in InfoCenter, use the following syntax:

```
/$INFOCENTER_WEBAPP_NAME/index?page=flow&key=$PROCESS_DOCUMENT_ID
```

You can populate ready-to-use, top-level process input variables in both SampleUI and `index?page=flow` through request parameters, and context variables through the following three methods, in decreasing precedence:

- Request parameter
- Request header
- Cookie

Troubleshooting

If a process flow is not found or there is a server-side error submitting a step, no error message appears.

The loading image appears indefinitely.

If the RuntimeUI server is not running, no error message appears.

The InfoCenter template with a blank content section appears.

Customizing the SampleUI

The SampleUI web application is provided in source code format at `<AnswerFlow installation directory>/Sample/SampleUI`. You have the option to modify this project and then run `deploy.sh` or `deploy.bat` in a RuntimeUI ICE in order to compile and deploy a customized SampleUI web application.

For example, you can modify Java files to change server-side functionality, modify Freemarker files to change generated HTML, or modify JavaScript files to change client-side functionality.

WARNING! If you add, modify, or delete any files under `<AnswerFlow installation directory>/Sample/SampleUI`, then you may need to redo customizations after patching or upgrading AnswerFlow, because the patch or upgrade script may revert your customizations or change SampleUI in a way that is incompatible with your customizations. After you apply the patch, you can find a backup of the pre-patched version of SampleUI under `Sample/SampleUI` in a zip file located at `<AnswerFlow installation directory>/backups`.

Requirement

The ability to modify the Sample UI source code.

About SampleUI Source Code

Within `<AnswerFlow installation directory>/Sample/SampleUI`, some of the important files and directories are listed in the following table.

File or Directory	Description
`SampleUI/build.properties`	Properties file used by SampleUI/build.xml
`SampleUI/build.xml`	Ant build file that compiles SampleUI. This is invoked by the deploy.sh/bat script
`SampleUI/lib/`	Contains jars used by the SampleUI project and AnswerFlow service calls
`SampleUI/src/main/java/`	Contains SampleUI's Java code
`SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/ContextParameterRetriever.java`	Retrieves context variable values at the beginning of each top-level process execution
`SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/MainController.java`	Java code for servicing HTTP requests to the following URLs: /SampleUI /SampleUI/dashboard /SampleUI/flow /SampleUI/start
`SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/ProcessClientsManager.java`	Handles creating new process execution contexts (interface name ProcessClient) and evicting expired ones (ready-to-use, the oldest ProcessClient is evicted once the HTTP session has a number of ProcessClients greater than the value of the configuration property common.maximumConcurrentlyExecutingProcessesPerSession)
`SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/ViewerController.java`	Superclass of MainController; services HTTP requests to /SampleUI/step
`SampleUI/src/main/java/com/inquira/pmeditor/security/SampleUserNameService.java`	Determines the user credentials to authenticate to Information Manager with (ready-to-use, the configuration properties sampleUi.credentials.username, sampleUi.credentials.password, and common.credentials.repository are used for the credentials)

File or Directory	Description

`SampleUI/src/main/resources/commonMessages_fr.properties`
`SampleUI/src/main/resources/commonMessages_ja.properties`
`SampleUI/src/main/resources/commonMessages.properties`
`SampleUI/src/main/resources/commonMessages_zh_CN.properties`
`SampleUI/src/main/resources/commonMessages_zh_TW.properties`
`SampleUI/src/main/resources/sampleUIMessages_de.properties`
`SampleUI/src/main/resources/sampleUIMessages_fr.properties`
`SampleUI/src/main/resources/sampleUIMessages_ja.properties`
`SampleUI/src/main/resources/sampleUIMessages.properties`
`SampleUI/src/main/resources/sampleUIMessages_zh_CN.properties`
`SampleUI/src/main/resources/sampleUIMessages_zh_TW.properties`
	Localized messages intended for display to an end-user

`SampleUI/src/main/resources/log4j.xml`	Log4j configuration
`SampleUI/src/main/resources/ok_jms.properties`	Oracle Knowledge Analytics configuration
`SampleUI/src/main/webapp/error.jsp`	Generic error page
`SampleUI/src/main/webapp/resources/css/`	CSS files to style the dashboard and process execution pages

`SampleUI/src/main/webapp/resources/css/preview/common.css`
`SampleUI/src/main/webapp/resources/css/preview/layout.css`
`SampleUI/src/main/webapp/resources/css/preview/loq.css`
	CSS files to style the dashboard and process execution pages

`SampleUI/src/main/webapp/resources/images/`	Image files
`SampleUI/src/main/webapp/resources/js/`	JavaScript files
`SampleUI/src/main/webapp/resources/js/preview/panelBuilder.js` `SampleUI/src/main/webapp/resources/js/preview/renderProcess.js`	JavaScript files that invoke the server to execute a process and render the result returned by the server
`SampleUI/src/main/webapp/WEB-INF/applicationContext.xml`	Spring application context configuration
`SampleUI/src/main/webapp/WEB-INF/SampleUI-servlet.xml`	Spring webapp context configuration

File or Directory	Description
`SampleUI/src/main/webapp/WEB-INF/site/`	Freemarker templates
`SampleUI/src/main/webapp/WEB-INF/site/localizedJs.ftl`	Freemarker template for JavaScript that needs server-side values (for example, localized messages)
`SampleUI/src/main/webapp/WEB-INF/site/preview/dashboard.ftl`	Freemarker template for the HTML rendered at /SampleUI/dashboard
`SampleUI/src/main/webapp/WEB-INF/site/preview/previewProcess.ftl`	Freemarker template for the HTML rendered at /SampleUI/flow
`SampleUI/src/main/webapp/WEB-INF/weblogic.xml`	WebLogic deployment descriptor (this file is ignored if you are not using a WebLogic application server)
`SampleUI/src/main/webapp/WEB-INF/web.xml`	Java web application deployment descriptor

About Functional Mappings in SampleUI

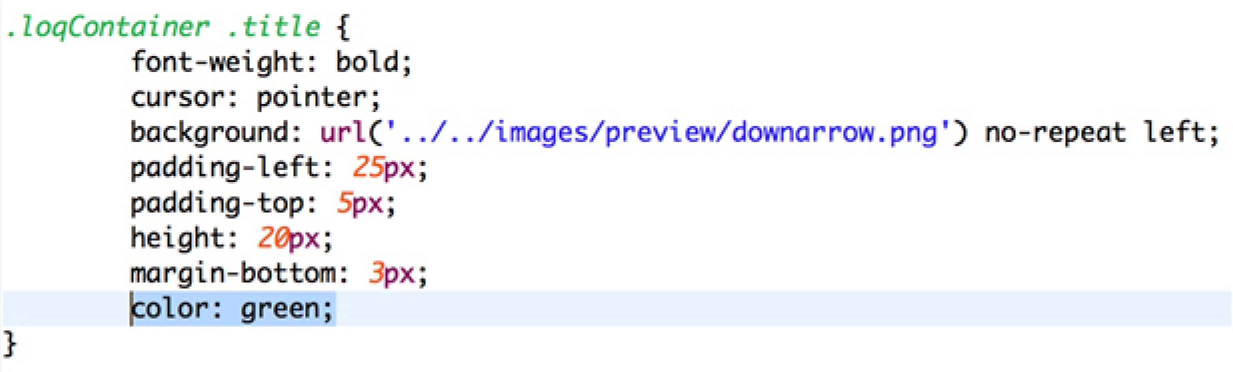
The following table below lists code references that implement functionality in SampleUI.

URL	Description	Source location
`/SampleUI`	Redirects to /SampleUI/dashboard	The method "base" in SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/MainController.java
`/SampleUI/dashboard`	A listing of all promoted processes, including a category tree for filtering and tabs to indicate whether to execute the published or latest version of the process document	• The method "dashboard" in SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/MainController.java populates the lists of published and latest processes and the category tree • The file SampleUI/src/main/webapp/WEB-INF/site/preview/dashboard.ftl renders the page's HTML
`/SampleUI/flow HTML`	page for executing a process	• The method "primaryView" in SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/MainController.java fetches the process for execution • The file SampleUI/src/main/webapp/WEB-INF/site/preview/previewProcess.ftl renders the page's initial HTML • The file SampleUI/src/main/webapp/resources/js/preview/renderProcess.js invokes the server to execute the process and renders the result returned by the server
`/SampleUI/start JSON`	API endpoint for starting process execution	The method "start" in SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/MainController.java
`/SampleUI/step JSON`	API endpoint for submitting a step during process execution	The method "step" in SampleUI/src/main/java/com/inquira/pmeditor/controller/spring/ViewerController.java

Reviewing a Customization Example

The following example shows how to customize the SampleUI code to change the color of the page titles in SampleUI from black to green.

1 Edit `SampleUI/src/main/webapp/resources/css/preview/loq.css` to add color: green; under the selector for `.loqContainer.title`, as shown in the following figure.



```
.loqContainer .title {
        font-weight: bold;
        cursor: pointer;
        background: url('../../images/preview/downarrow.png') no-repeat left;
        padding-left: 25px;
        padding-top: 5px;
        height: 20px;
        margin-bottom: 3px;
        color: green;
}
```

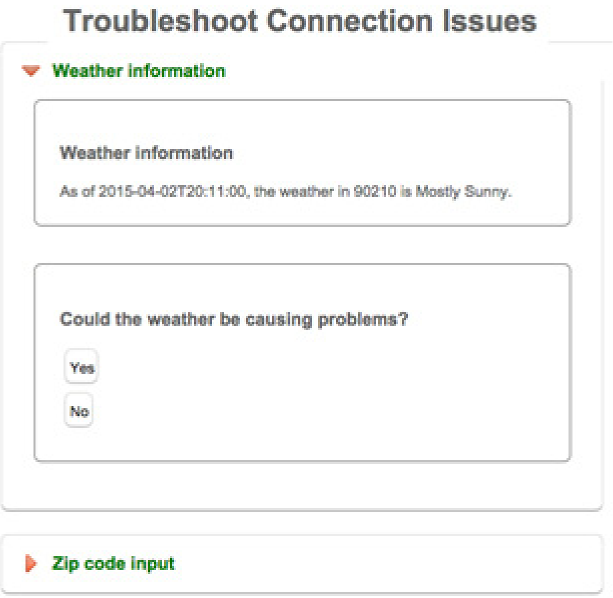
2 Compile and deploy SampleUI by running `deploy.sh` (or `deploy.bat` on Windows) within a RuntimeUI ICE.

3 Ensure the RuntimeUI server is running.

4 Navigate to `http://<SampleUI host>:<SampleUI port>/SampleUI`.

5 Launch a process.

6 Verify that the page titles are now green, as shown in the screenshot below.



Troubleshoot Connection Issues

Weather information

Weather information

As of 2015-04-02T20:11:00, the weather in 90210 is Mostly Sunny.

Could the weather be causing problems?

Zip code input

Integrating Process Flows in Web Applications by Using JSON

This section contains a description of SampleUI's JSON API.

You can deploy AnswerFlow to end users by having a front-end web application (such as Information Center or a different custom web application) render processes to the user, while the front-end web application communicates with SampleUI through SampleUI's JSON API. SampleUI then does the back end data processing that represents executing the processes, such as stepping through nodes and performing service calls.

Note: Unless otherwise specified, all human-readable messages built in to AnswerFlow that are returned in these responses are translated into the locale indicated by the request header Content-Language.

Requirements

Understand how the JSON API executes processes and submits requests, along with an understanding of the formats the JSON API uses. An example of using the JSON API ends this section, along with information about JSONP.

The JSON API supports executing processes. It does not currently support listing processes that can be executed, so you would not be able to implement something like SampleUI's dashboard with the ready-to-use JSON API. However, you can customize SampleUI to add more JSON-returning URLs if necessary.

To start executing a process, send a request to `/SampleUI/start`. The process continues executing until an error occurs, execution reaches a page, or execution reaches the end of the process. When process execution reaches a page, send a request to `/SampleUI/step` to submit the page. Within the `/SampleUI/step` request, the process continues executing until an error occurs, execution reaches a page, or execution reaches the end of the process. Continue to send requests to `/SampleUI/step` until the process ends or you wish to stop executing it.

Executing Processes with /SampleUI/start

Invoking this URL starts executing a process that continues until an error occurs, execution reaches a page, or execution reaches the end of the process.

HTTP Method: GET or POST

Either the GET command or POST command execute the same process.

PARAMETERS

- key: The document ID of the process to execute.
- latest (optional; default = "false"): If "true", then the latest version of the process document is executed. If “false", then the published version of the process document is executed.
- sessionId (do not pass in this parameter): This is used for SampleUI-internal usage. External applications should not use this parameter with the JSON API.
- Any request parameters, including the above and any others, that are passed in can be used to populate input and context variables (see below).

For each local variable marked as "Is Input?" in the top-level process, if a request parameter of the same name is passed in, then the local variable is set to the value of that parameter. If a request parameter of the same name is not passed in, then the local variable is set to its initial value.

For each context variable used within the top-level process or subprocesses, the following are checked for existence (in decreasing order of preference):

- Request parameter of the same name as the context variable
- Request header of the same name as the context variable
- Request cookie of the same name as the context variable

If any of these requests exists, then the value of the one of highest precedence is used as the value of the context variable throughout the top-level process and subprocesses. If none of these exist, the context variable's value is an empty string.

RESPONSE: SEE RESPONSE FORMATS, BELOW

SampleUI associates the data for each process execution session with the Java HTTP session. Therefore, if the response from `/SampleUI/start` sets the JSESSIONID cookie, then subsequent requests to `/SampleUI/step` within the same process execution session must pass the JSESSIONID cookie in the request. There can be multiple process execution sessions per HTTP session. If an HTTP session contains a number of process executions greater than or equal to the value of the configuration parameter `common.maximumConcurrentlyExecutingProcessesPerSession`, then the next time a process starts executing in that HTTP session, the oldest process execution in that HTTP session becomes invalid, and any requests to `/SampleUI/step` for the invalid process execution session will fail with "status": "error".

Submitting a Form Within a Process by Using /SampleUI/step

Invoking this URL submits a form within the process. The process continue executing until an error occurs, execution reaches a page, or execution reaches the end of the process.

HTTP Method: GET or POST

Either the GET command or POST command execute the same process.

PARAMETERS

- sessionId: The sessionId returned from /SampleUI/start.
- stepId: The stepId of the step that contains the submitted form. This can be the current step or any prior step. When a step before the current one is submitted, process execution backtracks to that step (except that local variables remain at their current values), and any steps beyond the submitted one can no longer be submitted.
- displayFormReferenceKey: The referenceKey of the submitted form.
- Any request parameter whose name is the same as the datasource of one of the submitted form's form controls is treated as the submitted value of that form control. For Check Box and Multi-Select form controls, if the user selects multiple values, then the form control's datasource should be populated as a separate request parameter for each value. For example, if the user selects Check Boxes "A" and "B" and the datasource is "d16e276c-c765-11e4-9468-58b035f3c92b", then in a GET request to `/SampleUI/step`, the query string should contain "d16e276c-c765-11e4-9468-58b035f3c92b=A&d16e276c-c765-11e4-9468-58b035f3c92b=B".

- These request parameters populate the Input Variables of their respective form controls.

FORMATS

Nine formats are documented below: response, issue, page element, form control, form control type, Information Manager document, Information Manager document attribute, Information Manager document attribute type, and Information Manager document attribute content.

RESPONSE

`/SampleUI/start` and `/SampleUI/step` each return one of the following four types of responses:

- The process execution has reached a page.

```
    {
    "status": "success",
    "sessionId": <Process execution session ID (string)>,
    "stepId": <Step ID (The step ID is different every time execution reaches a page,
    even if the page is visited multiple times through e.g. a loop of nodes) (string)>,
    "errors": <Errors (array of issues)>,
    "warnings": <Warnings (array of issues)>,
    "panel": {
       "title": <Page title (string)>,
       "pageElements": <Page elements (array of page elements)>
  }
}
```

- The process execution has ended.

```
{
   "status": "success",
   "sessionId": <Process execution session ID (string)>,
   "stepId": <Step ID (string)>,
   "errors": <Errors (array of issues)>,
   "warnings": <Warnings (array of issues)>,
   "outputVariableEntries": <Output variable entries (i.e. variables in the top-level
   process marked as "Is Output?") (array of variable entries)>
}
```

- An expected error has occurred.

```
    {
    "status": "error",
    "errorMessage": {
       "titleText": <Plain text (i.e. not HTML) error message title (string)>,
       "bodyHtml": <HTML error message body (string)>
  }
}
```

- An unexpected error has occurred.

An unexpected error could include, for instance, a missing request parameter, or a bug in the SampleUI code. The response code indicates an error. The response body is an HTML error page.

Issue:

```
{
   "messageCode": <Machine-readable identifier indicating the nature of the issue (string)>,
   "replacementValues": <Contextual data relevant to the specific instance of the issue
   (array of strings)>,
   "localizedMessage": <Human-readable error message (string)>
}
```

EXAMPLE ISSUE

```
{
    "messageCode": "sampleUi.document.noMasterIdentifier",
    "replacementValues": [
    "AF1"
    ],
    "localizedMessage": "Document AF1 has no master identifier. Please check the document's
    channel schema page to confirm the attribute noted as the master identifier; then confirm
    you have entered information into the document field labeled in the schema as the master
    identifier."
}
```

PAGE ELEMENT

There are three types of page element formats:

- Display form element

```
{
    "label": <Page element label (string)>,
    "type": "form",
    "referenceKey": <Form reference key (string)>,
    "formControls": <Form controls (array of form controls)>
}
```

- Display HTML element

```
{
      "label": <Page element label (string)>,
      "type": "rawData",
      "html": <HTML to display (string)>
}
```

- Display Information Manager documents element

```
{
    "label": <Page element label (string)>,
    "type": "docReferences",
    "imDocuments": <Information Manager document (array of Information Manager documents)>
}
```

FORM CONTROL

```
{
    "type": <Form control type (form control type)>,
    "label": <Label (string)>,
    "className": <Class name (string)>,
    "toolTip": <Tooltip (string)>,
    "datasource": <The reference key (a UUID) of the local variable assigned as this form
    control's Input Variable (string)>,
    "defaultValue": <Default value for Hidden, Password, Radio, Select, Text, and Text Area
    (string)>,
    "ghostedText": <Ghost text (string)>,
    "url": <Image URL (string)>,
    "required": <Required (boolean)>,
    "options": <Options (array of strings)>,
    "defaultValues": <Default values for Check Box and Multi-Select (array of strings)>
}
```

Note: Not all of these properties are applicable to all form control types.

FORM CONTROL TYPE

Form control type is one of the following:

JSON type	Type displayed in AnswerFlow Editor
"checkbox”	Check Box
"hidden”	Hidden
"label”	Label
"multiSelect”	Multi-Select
"password”	Password
"radio”	Radio
"select”	Select
"submit”	Submit
"text”	Text
"textArea”	Text Area

Information Manager Document

There are two types of Information Manager document formats:

- Successfully retrieved document

```
{
   "attributes": [{
      "label": <Channel name (string)>,
      "children": <Top-level attributes (array of Information Manager document attributes)>
   }]
}
```

- Unsuccessfully retrieved document

```
{
   "attributes": [{
   "content": <Document retrieval error message (string)>,
   "type": "EXCEPTION"
   }]
}
```

Information Manager Document Attribute

There are two types of Information Manager document attribute formats:

- Node

```
{
      "label": <Node name (string)>,
      "children": <Child attributes (array of Information Manager document attributes)>
}
```

- Attribute

```
{
   "label": <Attribute name (string)>,
   "content": <Attribute content (Information Manager document attribute content)>,
   "listContent": <Array of HTML strings representing list entries (only for attributes of
   type List) (note that out of the box, SampleUI renders these strings as plaintext rather
   than HTML due to "Bug 20715704 - Lists in IM documents render as text instead of HTML")
   (array of strings)>,
   "type": <Attribute type (Information Manager document attribute type)>,
   "url": <File URL (only for attributes of type File) (string)>
}
```

Information Manager Document Attribute Type

Attribute type is one of the following:

JSON Type	Type Displayed in Information Manager Console
"BOOLEAN”	Check Box
"COUNTER”	Counter
"DATE”	Date
"DATETIME”	Date Time
"FILE”	File
"FLOAT”	Float
"INTEGER”	Integer
"LIST”	List
"TEXT_AREA”	Text Area
"TEXT_FIELD”	Text Field
"TIME”	Time
"WYSIWYG_EDIT”	Rich Text Area

Information Manager Document Attribute Content

Attribute content is a string in the following type-specific formats:

Type displayed in Information Manager console	Attribute content format
Check Box	Checked check boxes have the content "Y", but unchecked check boxes do not appear in the JSON at all.
Date	A timestamp in the format "<yyyy>-<MM>-<dd> <HH>:<mm>:<ss> Etc/GMT". The date entered in the Information Manager console is interpreted as midnight on the entered date in the Information Manager server's time zone. The GMT timestamp returned here represents the same date and time, but in the GMT time zone. For example, if the Information Manager server is in the PST time zone (GMT-8), then a Date entered as "03/12/2015" in the Information Manager console would render as "2015-03-12 08:00:00 Etc/GMT" in the JSON.
Date Time	A timestamp in the format "<yyyy>-<MM>-<dd> <HH>:<mm>:<ss> Etc/GMT". The date and time entered in the Information Manager console is interpreted as relative to the Information Manager server's time zone. The GMT timestamp returned here represents the same date and time, but in the GMT time zone. For example, if the Information Manager server is in the PST time zone (GMT-8), then a Date Time entered as "03/12/2015 06:00 AM" in the Information Manager console would render as "2015-03-12 14:00:00 Etc/GMT" in the JSON.
File	File name
List	null

Type displayed in Information Manager console	Attribute content format
Time	A timestamp in the format "<yyyy>-<MM>-<dd> <HH>:<mm>:<ss> Etc/GMT". The time entered in the Information Manager console is interpreted as the entered time on 1970-01-01 (i.e. the Unix Epoch), relative to the Information Manager server's time zone. The GMT timestamp returned here represents the same date and time, but in the GMT time zone. For example, if the Information Manager server is in the PST time zone (GMT-8), then a Time entered as "06:00 AM" in the Information Manager console would render as "1970-01-01 14:00:00 Etc/GMT" in the JSON.
All other types	HTML representing the content stored in the attribute in the Information Manager document

VARIABLE ENTRY

```
{
    "name": <Variable name (string)>,
    "value": <Variable value (string)>
}
```

Example: The following example demonstrates a sequence of requests and responses using SampleUI's JSON API to execute the process Troubleshoot Connection Issues, which is available ready-to-use in AnswerFlow as an example process. For the purposes of the example, assume that Troubleshoot Connection Issues is published as the document AF1, which is not the case ready-to-use.

1 Start process execution:

```
Request: GET /SampleUI/start?key=AF1
```

RESPONSE

```
{
    "status": "success",
    "sessionId": "8b437e14-5be5-412a-aaa6-862a191956f2",
    "stepId": "GUID-2-Troubleshoot Connection Issues",
    "errors": [],
    "warnings": [],
    "panel": {
        "title": "Zip code input",
        "pageElements": [{
            "label": "What is the customer's zip code?",
            "type": "form",
            "referenceKey": "9b315382-47ac-4d39-b44b-38dbe0e98960",
            "formControls": [{
                "type": "text",
                "label": "Zip code",
                "className": "",
                "toolTip": "",
                "datasource": "64852e35-ccbd-408d-99a5-bc20e7fc3e93",
                "defaultValue": "",
                "ghostedText": "",
                "url": "",
                "required": true,
                "options": [],
                "defaultValues": null
            }]
        }]
    }
}
```

2 Submit the value "90210" for the field with label "Zip code":

```
Request: GET /SampleUI/step?sessionId=8b437e14-5be5-412a-aaa6-862a191956f2&stepId=
GUID-2-Troubleshoot+Connection+Issues&displayFormReferenceKey=9b315382-47ac-4d39-
b44b-38dbe0e98960&64852e35-ccbd-408d-99a5-bc20e7fc3e93=90210
```

RESPONSE

```
{
    "status": "success",
    "sessionId": "8b437e14-5be5-412a-aaa6-862a191956f2",
    "stepId": "GUID-5-Troubleshoot Connection Issues",
    "errors": [],
    "warnings": [],
    "panel": {
        "title": "Weather information",
        "pageElements": [{
            "label": "Weather information",
            "type": "rawData",
            "html": "As of 2015-03-17T01:02:00, the weather in 90210 is Partly Cloudy."
            }, {
            "label": "Could the weather be causing problems?",
            "type": "form",
            "referenceKey": "f915f2d9-fe06-49fe-95f4-92d9263f9b7e",
            "formControls": [{
                "type": "submit",
                "label": "Yes",
                "className": "",
                "toolTip": "",
                "datasource": "670c9826-05cd-4113-a241-0a1c0fdc48b2",
                "defaultValue": "",
                "ghostedText": "",
                "url": "",
                "required": false,
                "options": [],
                "defaultValues": null
        }, {
                "type": "submit",
                "label": "No",
                "className": "",
                "toolTip": "",
                "datasource": "670c9826-05cd-4113-a241-0a1c0fdc48b2",
                "defaultValue": "",
                "ghostedText": "",
                "url": "",
                "required": false,
                "options": [],
                "defaultValues": null
            }]
        }]
    }
}
```

3 Submit the button with the label "Yes":

Request: GET /SampleUI/step?sessionId=8b437e14-5be5-412a-aaa6-862a191956f2&stepId=GUID-5-Troubleshoot+Connection+Issues&displayFormReferenceKey=f915f2d9-fe06-49fe-95f4-92d9263f9b7e&670c9826-05cd-4113-a241-0a1c0fdc48b2=Yes

RESPONSE

```
{
    "status": "success",
    "sessionId": "8b437e14-5be5-412a-aaa6-862a191956f2",
    "stepId": "GUID-7-Troubleshoot Connection Issues",
    "errors": [],
    "warnings": [],
    "panel": {
       "title": "Solution",
       "pageElements": [{
          "label": "Weather",
          "type": "rawData",
          "html": "Wait for the weather to clear up."
       }]
    }
}
```

JSONP

If you pass the request parameter "callback" to these URLs, then instead of JSON, they return JSONP, with the value of "callback" used as the JSONP function name. SampleUI's ready-to-use JavaScript code uses the JSONP interface.

Example: A GET request to /SampleUI/start?key=AF2&callback=jsonpCallback might return the following:

```
jsonpCallback({
   "warnings": [],
   "stepId": "GUID-2-31CDD9FA7DEB4DFB97DB899F977C3608",
   "sessionId": "9d419f04-845d-488d-b6c5-642277e25d72",
   "panel": {
      "title": "Page",
      "pageElements":[}
         "label": "Display HTML",
         "html": "Hello, World!",
         "type": "rawData"
      }]
   ]},
   "errors": [],
   "status": "success"
})
```

Using Search for Process Integration

You can integrate process flows in web applications by using Search to:

- Return a search result for an AnswerFlow process flow by way of a Managed Answer or by firing an Intent response.
- Crawl the AnswerFlow Information Manager Channel and find process flows by way of Search Results.

Searching for Process Flows by Using Managed Answers

This topic explains how to use Managed Answers to find process flows.

Requirements

- Know how to use Oracle Knowledge Language Workbench Dictionary Manager
- A published AnswerFlow process flow URL

Create a Managed Answer rule in Dictionary Manager using the Custom Content method. Link the Managed Answer to an AnswerFlow process by using one of the following URLs (which launch the process) as the URL of the Managed Answer:

- SampleUI URL, for example, `http://example.com:8233/SampleUI/flow?key=DOC1`
- Information Center URL, for example `http://example.com:8226/infocenter/index?page=flow&key=DOC1`

Note: Creating a Managed Answer in the Language Workbench Dictionary Manager application is not a trivial task; if you are not familiar with Dictionary Manager, please refer to Oracle Knowledge Intelligent Search Language Tuning Guide for more information on configuring Managed Answers.

Using Search Results to Find Process Flows in the AF Channel

To use search results to locate process flows in the AnswerFlow Channel, you must meet the following requirements:

- Know how to use Oracle Knowledge Information Manager and System Manager
- Ability to define and run an Internal Collection Crawl in System Manager
- Add to AnswerFlow channel schema human-readable attributes such as Process title and abstract

Note: The procedures for creating an Internal Collection crawl in the System Manager are documented in *Oracle Knowledge Intelligent Search Administration Guide*.

Using System Manager, set up an internal crawl of the AnswerFlow Information Manager channel. When configuring this crawl, provide the following settings, depending on whether your users will execute processes directly in SampleUI or in InfoCenter from the search results returned.

Using Search Results to Launch Process Flows

You can launch process flows in the SampleUI or in Information Center.

Launching Process Flows in the SampleUI

To direct the AnswerFlow document search result to launch the process in SampleUI, configure the Information Manager Collection in the Search System Manager with the following settings:

```
Protocol: <SampleUI protocol>
Host: <SampleUI hostname>
Port: <SampleUI port>
Prefix: SampleUI/flow?key=
Suffix: <blank>
```

Launching Process Flows in Information Center

To direct the AnswerFlow document search result to launch the process in Information Center (assuming page=flow is configured according to the customization instructions), configure the Information Manager Collection in the Search System Manager with the following settings:

```
Protocol: <InfoCenter protocol>
Host: <InfoCenter hostname>
Port: <InfoCenter port>
Prefix: <InfoCenter context path>/index?page=flow&key=
Suffix: <blank>
```