



# Oracle Knowledge Intelligent Search Optimization Guide

---

*Fully Utilizing Language Workbench Options*

---

Oracle Knowledge Version 8.5  
Document Number OKIS-OPT85-01  
January 14, 2013

**Oracle, Inc.**

## COPYRIGHT INFORMATION

Copyright © 2002, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

# Contents

---

<b>Preface</b>	<b>About This Guide</b>	<b>1</b>
	In This Guide	1
	Screen and Text Representations	1
	References to World Wide Web Resources	1
<b>Chapter 1</b>	<b>Dictionary Manager Advanced Features</b>	<b>3</b>
	Advanced Features of Concepts	4
	Indexed Concept	4
	Openclass	4
	Subsumes	4
	Advanced Features of Rules	5
	Add Business Conditions	5
	Add Business Conditions	6
	Add Answer Action	7
	Add Search Component	7
	Add Facet Restriction	7
	Add Set Attribute	7
	Add Relative Weights	8
	Add Scope Weights	9
	Add Suppress Rules	10
	Answer Action Methods for Rules	11
	Specific Excerpt Answer Action Method	12
	Search Answer Action Method	12
	Structured Query Answer Action Method	13
	Glossary Answer Action Method	14
	Plug-in Answer Action Method	15
	Process Wizard Answer Action Method	16
	Alias Lists	16
	Create a New Alias List	16
	Create a Rule to Fire When an Alias List Key is Matched	18
	Pass User-Selected Facets Through an Alias List	18

Alias List Tools	18
Import Text Files	18
Validate Alias Lists	19
Advanced Feature of Alias Lists	19
Set Overriding	19
Exception Lists	19
Edit an Exception List	21
Create a New Exception List	22
Exception List Tools	23
Spell Check Lists	23
Edit a Spell Check List	24
Create a New Spell Check List	25
Spell Check List Tools	26
Intent Hierarchy	26
Create a new Intent Hierarchy	27
User Profiles	28
Create a new User Profile	29
User Profile Tools	29
Validate Profile	30
Show Concepts Tree	30
Subject Maps	30
Edit Subject Maps	32
Change the Default Subject Map	33
Oracle Knowledge Language Analysis	33
Language Analysis Components	33
The Language Analysis Process	34
The Content Store and Indexes	34
Analyzing Requests	34
Evaluating Requests	35
Evaluating Spelling	35
Evaluating Business Conditions	35
Evaluating Language Conditions	35
Defining Response Criteria	36
Types of Actions	36
Evaluating Actions	36
The Scoring Process	36
Search Component Scores	37
The Document Relevance and Recency Scores	37
Evaluating Anaphora	38
Creating the Saliency List	38
Identifying Anaphoric Expressions	39
Determining the Head of the Anaphoric Expression	39
Testing Referring Expressions Within the Current Request	40
Resolving Anaphoric Expressions	40
Annotating the Anaphoric Expression	40
<b>Chapter 2 Process Wizards</b>	<b>41</b>
Implementing the Process Wizard Feature	42
Process Wizard Elements	42

Process Wizard Steps .....	43
Process Wizard Fields .....	43
Process Wizard Conditions and Transitions .....	43
Process Wizard User Interface Elements .....	44
Accessing the Process Wizard Editor .....	44
Using the Process Wizard Editor .....	46
The Graphic View .....	47
Step Displays in the Graphic View .....	48
Process Wizard Editor Step Identification .....	49
Conditions and Transitions in the Graphic View .....	49
The Process Wizard Inspector .....	50
The Default Step .....	51
The Process Step Inspector .....	51
The Step Fields Tab .....	53
The Step Transitions Tab .....	54
The Step Preview Tab .....	55
The Shared Fields Inspector .....	56
Tutorial: Creating a Process Wizard .....	57
Scenario: Creating a Process to Assist Message Retrieval .....	57
Designing the Wizard .....	58
Designing the Process Steps .....	59
Designing Step Fields .....	59
Designing Step Conditions and Transitions .....	60
Defining the Process Using the Process Wizard Editor .....	62
Defining a New Process .....	62
Defining the Initial Step .....	64
Defining the Introductory Text Field .....	65
Defining the User Options for the Initial Step .....	65
Previewing the Initial Step .....	67
Defining Information Steps .....	68
Defining the Email Message Location Step (Step 2A) .....	68
Creating the Email Location Step .....	69
Specifying the Condition for the Email Location Step .....	69
Specifying the Email Location Step Name .....	70
Defining the Introductory Text Field for the Email Location Step .....	71
Defining User Options for the Email Location Step .....	72
Defining Solution Steps .....	73
Defining the Local Email Solution Step .....	73
Creating the Local Email Solution Step .....	73
Specifying the Condition for the Local Email Solution Step .....	73
Defining the Text Field for the Local Email Solution Step .....	74
Tutorial Review .....	76
Creating Process Wizards .....	77
Creating New Process Wizards .....	77
Defining a New Process Wizard .....	77
Copying an Existing Process Wizard .....	78
Creating Process Steps .....	79
Creating a New First Step .....	80
Designating a New First Step .....	81
Deleting a Process Step .....	81
Unused Steps .....	82

Previewing Steps .....	82
Previewing Steps within the Process Step Inspector .....	82
Previewing Steps within a Browser .....	83
Defining Fields .....	84
Specifying Field Types .....	86
Specifying a Search Action within a Step Field .....	89
Creating Shared Fields .....	89
Creating a New Shared Field from an Existing Shared Field .....	91
Adding a Shared Field to a Process Step .....	91
Using a Shared Field as a Template .....	92
Creating a Shared Field from a non-Shared Field .....	92
Defining Conditions .....	93
Specifying Condition Field Names .....	93
Specifying Condition Values .....	93
Defining Transitions .....	94
Creating Dictionary Rules to Invoke Process Wizards .....	95
Assigning the Process Wizard Action to a Rule .....	95
Creating a Process Wizard Intent Response .....	96
The Process Wizard User Interface .....	96
The Process Wizard Answer .....	97
The Step Display Area .....	97
Modifying the Process Wizard User Interface .....	98

## **Chapter 3 Personalized Navigation Setup ..... 99**

Personalized Navigation Presentation .....	100
Personalized Navigation Implementation .....	100
The Personalized Navigation Application .....	101
Personalized Navigation Application Components .....	101
Taxonomies .....	101
Facets .....	102
Classifiers .....	103
Taxonomy Builders .....	103
Formatters .....	103
Comparators .....	104
Implementing Personalized Navigation .....	104
Activating the Personalized Navigation User Interface Layout .....	104
Accessing the Personalized Navigation Application .....	104
Defining a Navigation Application .....	105
Configuring the Navigation Application .....	106
Defining Taxonomies .....	107
Defining Taxonomy Display Properties .....	108
Specifying Automatic Facet Creation .....	111
Specifying the Content Source for the Taxonomy .....	112
Specifying Generation Options for Structured Content .....	113
Automatically Generating Facets For Database Content .....	113
Generating a Date Taxonomy for a Selected Column .....	114
Generating a Numeric Range (Price) Taxonomy .....	115
Generating a Facet Based on a Single Column .....	116
Generating a Taxonomy Based on Multiple Columns .....	117

Specifying Generation Options for Unstructured Content .....	117
Generating a Taxonomy Based on Directory Structure .....	118
Generating a Taxonomy from Concepts .....	118
Generating a Taxonomy from Meta-Tags .....	119
Configuring the Preprocessor to Preserve Meta Tags for Classification ..	120
Manually Assigning Database Content to a Facet .....	121
Assigning All Table Entries to a Facet .....	121
Assigning Selected Tables to a Selected Facet .....	121
Manually Assigning Document Content to a Facet .....	123
Assigning Documents to a Facet by Collection .....	123
Assigning Documents to a Facet by Type .....	124
Assigning Documents to a Facet by Language .....	125
Assigning Documents to a Taxonomy by Content .....	125
Assigning Documents to a Facet by URL .....	126
Performing Data Classification .....	127
Referring to Facets from Dictionary Rules .....	128
Referring to Facets by Name .....	128
Selecting Multiple Facets from the Tree Display .....	129
Selecting Multiple Facets using IML .....	129
Using VIL in Rules that Select Content Categories .....	130
Restricting the Display for Facets Above a Selected Facet .....	130
Preventing a Facet from Displaying .....	130
Specifying Approximate Facet Matching .....	130
Selecting a Range of Content Categories .....	131
Responding to Requests for Non-Existent Content Categories .....	131
Facet expansion in addition to restriction .....	131
Understanding Facet Counts .....	132
Using Advanced Facet Definition Settings .....	133
Specifying Advanced Facet Display Settings .....	135
Using Advanced Data Classification .....	138
Using an SQL Condition to Classify Content .....	139
Classifying Content Based on Database Column Values .....	140
Configuring Multiple Taxonomy levels from Structured Columns .....	141
Configuring the Appearance of Structured Column Child Facet Names .....	141
Example: Creating a Taxonomy from Database Categories .....	141
Example: Creating a Taxonomy from Dates in a Database .....	143
Example: Creating a Price Taxonomy from a Database .....	144
Classifying Large Taxonomies by Accessing the Data at Runtime .....	145
Classifying Content by Collection .....	146
Classifying Content by Document Type .....	147
Using IML Expressions to Classify Content .....	148
Classifying Content Based on Language .....	148
Classifying Content Based on HTML Meta Tags .....	149
Example: Creating a Taxonomy from Document Metadata .....	150
Classifying Content Based on Dictionary Concepts .....	151
Classifying Content by Directory Structure .....	152
Classifying Content by URL .....	153
Using the Unstructured Content Classifier .....	154
Personalized Navigation Application Classification Components .....	155
Structured Data Classifiers .....	155
The SQL Classifier .....	155

The Structured Column Classifier . . . . .	156
The Ad Hoc Column Classifier . . . . .	157
Ad Hoc Column Classifier Settings . . . . .	157
Unstructured Data Classifiers . . . . .	158
The Collection Classifier . . . . .	159
The Document Type Classifier . . . . .	160
The Meta Tag Classifier . . . . .	160
The Ontology Classifier . . . . .	161
The Site Graph Classifier . . . . .	162
The URL Classifier . . . . .	163
The Unstructured Content Classifier . . . . .	163
The IML Classifier . . . . .	164
The Language Classifier . . . . .	165
Advanced Data Classifier . . . . .	165
The Template Classifier . . . . .	165
IML Template Settings . . . . .	166
On-the-fly Range Template Settings . . . . .	166
Structured To Unstructured Template Settings . . . . .	168
URL Template Settings . . . . .	168
Personalized Navigation Application Taxonomy Builders . . . . .	169
The String Taxonomy Builder . . . . .	170
The Date Taxonomy Builder . . . . .	170
The Directory Taxonomy Builder . . . . .	172
The Range Taxonomy Builder . . . . .	173
Specifying Regular Expression Substitutions . . . . .	175
Personalized Navigation Application Comparators . . . . .	175
Personalized Navigation Application Formatters . . . . .	176
 <b>Chapter 4 User Manager . . . . .</b>	 <b>177</b>
Using User Manager . . . . .	178
Add New User . . . . .	179
User Permissions . . . . .	180
Modify a User Account . . . . .	180
Delete a User Account . . . . .	180
 <b>Appendix A Configuring Access to Business Data . . . . .</b>	 <b>181</b>
The Preference Service . . . . .	181
Access to Preference Data . . . . .	181
Hierarchical Namespace Traversal . . . . .	182
Preference Types . . . . .	182
Business Data Providers . . . . .	182
The Session Provider . . . . .	182
Session Provider Preferences . . . . .	183
Creating Custom Providers . . . . .	183
Defining a Provider in the Application . . . . .	184
Creating Custom Preferences . . . . .	184
Configuring Preferences . . . . .	185
Editing Preferences . . . . .	186
The Context Variables Page . . . . .	186
Displaying the Editable Context Variable List . . . . .	186
Selecting a Context Variable for Editing . . . . .	187



Adding and Modifying Context Variable Preference Parameters .....	187
The Preference API .....	188
The getPreference Method .....	188
The getExternalString Method .....	188
The getExternalCode Method .....	189
The getPreferenceValue Method .....	189
The SetPreferenceValue Methods .....	189
The rundown Method .....	191
The getPreferences Method .....	192
The getPreferenceKeys Method .....	192
The Provider API .....	193
The Provider Default Constructor .....	193
The getType Method .....	193
The get Methods .....	194
The set Method .....	195
The toString Method .....	195
The Provider rundown Method .....	195



# About This Guide

This guide is intended to be used by those already familiar with the features covered in the [Intelligent Search Language Tuning Guide](#), and want to use the Oracle Knowledge Language Workbench to further optimize the features of Oracle Knowledge Intelligent Search.

## In This Guide

The *Optimizing Oracle Knowledge Intelligent Search Guide* is divided into the following sections:

<a href="#">Chapter 1, Dictionary Manager Advanced Features</a>	Describes how to use the advanced features of the Dictionary Manager. See <a href="#">Chapter 2, Dictionary Manager</a> in the <a href="#">Intelligent Search Language Tuning Guide</a> for information how to use the basic features of the Dictionary Manager.
<a href="#">Chapter 2, Process Wizards</a>	Describes how to create Process Wizards that respond to specific questions with preconfigured dialogs.
<a href="#">Chapter 3, Personalized Navigation Setup</a>	Describes how to implement the Oracle Knowledge Personalized Navigation feature to categorize application content and display a hierarchy of the results, such as products and models, services, and prices, in the User Interface.
<a href="#">Chapter 4, User Manager</a>	Describes the Oracle Knowledge User Manager component accessed from within the Language Workbench that allows you to create and set permissions for Language Workbench users.
<a href="#">Appendix A, Configuring Access to Business Data</a>	Describes the Preference Service that accesses and stores business data items obtained from configured data sources.

## Screen and Text Representations

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.

## References to World Wide Web Resources

For your convenience, we refer 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.



# Dictionary Manager Advanced Features

The Oracle Knowledge Dictionary Manager is a component accessed from within the Language Workbench that allows you to customize and maintain the Dictionary Objects used by Oracle Knowledge Intelligent Search to return the best possible answers.

**Note:** Before using the information in this chapter, it is advised that you familiarize yourself with the [Intelligent Search Language Tuning Guide](#).

In addition to the [Advanced Features of Concepts on page 4](#) and [Advanced Features of Rules on page 5](#), this chapter describes the following Dictionary Objects:

- [Answer Action Methods for Rules on page 11](#)
- [Alias Lists on page 16](#)
- [Exception Lists on page 19](#)
- [Intent Hierarchy on page 23](#)
- [User Profiles on page 25](#)
- [Show Concepts Tree on page 27](#)
- [Subject Maps on page 28](#)

For information on Dictionary Objects not included, please see the [Intelligent Search Language Tuning Guide](#).

This chapter also describes:

- [Oracle Knowledge Language Analysis on page 30](#)
- [The Scoring Process on page 33](#)

## Advanced Features of Concepts

This section explains how to use the advanced features of Concepts. For more information on Concepts please see [Intelligent Search Language Tuning Guide](#)

Select **Advanced** from the Concept window. The following options are displayed:

- [Indexed Concept on page 4](#) (Checked by default.)
- [Openclass on page 4](#), (Unchecked by default.)
- [Subsumes on page 4](#) (Checked by default.)

### Indexed Concept

The **Indexed Concept** option located in the Advanced menu of the Concept window is checked by default. When the Indexed Concept option is checked, the System Manager reindexes the Concept upon running an indexing job. For more information on Indexing, please see [Oracle Knowledge Intelligent Search Administration Guide](#).

If the Index job is too slow, you may uncheck the Indexed Concept option. To uncheck the Indexed Concept option from multiple Concepts use the Dictionary Find Tool to find Concepts with Indexed Concept set.

Once all the Concepts with IML indexing are displayed:

- Select All Concepts.
- 'Right-click', and select **Unset Indexed Concept**.

### Openclass

The **Openclass** option located in the Advanced menu of the Concept window allows you to manually classify a Concept as openclass. This option is only applicable for Concepts that include a suitable IML pattern, such as those that end in "Openclass", like `date Openclass`. An *Openclass Concept* defines an open-ended class of strings which fit a particular pattern, but which do not have interchangeable meanings. Typical examples are email addresses and dates. Openclass Concepts must be defined with IML.

The Openclass option is unchecked by default<sup>1</sup>. Upon checking the Openclass option, the query containing the Openclass Concept is annotated with an #OPENCLASS label on the string. This #OPENCLASS label prevents Concept-related Search Component Rules (for example, Required Concept SC) from firing. Instead Oracle Knowledge Intelligent Search uses a set of specific Openclass Search Component Rules that match on the string-form representation, and do not match on synonyms or clusters. .

For example, a query containing the string "1/12/10" is annotated with the openclass <Date Openclass> Concept. This Concept triggers Oracle Knowledge Intelligent Search to find the numbers in a string, versus finding the numbers individually.

### Subsumes

The **Subsumes** option located in the Advanced menu of the Concept window is checked by default.

When the **Subsumes** option is checked, strings that match overlapping sets of synonyms will index only the longest match. If unchecked, strings that match overlapping sets of synonyms will index all matches.

---

1. One exception to this is the concept <swearword Openclass>, which includes a Synset of swear words for each language.

For example, if the Subsumes option is checked, the phrase "information manager" is indexed with only the Concept `<information manager>`. If the Subsumes option is unchecked, "information manager" is indexed with the Concepts `<information manager>`, `<information>`, and `<manager>`. Queries are always annotated with Subsumes on, the checkbox only affects document indexing

Another example, in a document with the sentence:

Click here to get started by opening a free checking account.

With `<free checking account>` and `<checking account>` concepts set to subsume, the phrase "free checking account" is indexed only with the concept `<free checking account>`. If there were no `<free checking account>` concept (or synonym) then it would be indexed with `<checking account>`, and only if there were no `<checking account>` concept it would be indexed with `<account>`.

With subsume turned off, the phrase "free checking account" would be indexed with the `<free checking account>` concept; "checking account" with the `<checking account>` concept; and "account" with the `<account>` concept. In either case, a user query with the phrase "free checking account" annotates the concept `<free checking account>`.

The result in these cases is that if subsume is on, and a user searches for "open an account", there would be a stem/string match but not a concept match on that document, resulting in a different score; and any custom search components that look for a concept annotation won't match. With subsume off, that document/sentence would be matched with both a stem/string match and the concept annotation.

## Advanced Features of Rules

This section explains how to use the advanced features of Rules. For more information on Rules please see [Intelligent Search Language Tuning Guide](#).

Select **Advanced** from the Rule window. The following options are displayed:

- [Add Business Conditions on page 5](#)
- [Add Answer Action on page 7](#)
- [Add Search Component on page 7](#)
- [Add Facet Restriction on page 7](#)
- [Add Set Attribute on page 7](#)
- [Add Relative Weights on page 8](#)
- [Add Scope Weights on page 9](#)
- [Add Suppress Rules on page 10](#)

This section also provides [Answer Action Methods for Rules on page 11](#).

## Add Business Conditions

*Business Conditions* enable you to tailor search results based on context, such as the user's identity, location, or date and time of the query. You can configure Business Conditions to access information from the user's current Oracle Knowledge session, as well as from external sources of user and session information.

Business Conditions make use of Context Variables. *Context Variables* are properties that correspond to configured preferences within Oracle Knowledge. With the exception of date- and time- parameters, you must configure providers for all preferences. For more information, please see [Chapter A, Configuring Access to Business Data](#).

## Add Business Conditions

To add a Business Condition:

- 1 Select **Add Business Conditions** from the **Advanced** menu of the Rule window. A Business Conditions tab displays:

- 2 Click **Add new condition**. The Business Condition window displays:

- 3 Select the Context Variable from the first drop down menu. For more information please see [The Context Variables Page on page 187](#).
- 4 Select a Condition Operator from the middle drop down menu. .
- 5 Enter the value that the Rules Engine uses to compare the current value of the context variable, as specified by the condition operator.
- 6 Click **Add a line**, if you want to add alternate Business Conditions to a Rule.
- 7 Click **Add new condition**, if you want to add multiple Business Conditions.

**Note:** Use **Add a line** to create a single Business Condition which is true if any of multiple variable comparisons are true. The rule is triggered only if *all* of the Business Conditions are *true*.

- 8 Click **OK**. The Business Condition appears in the Business Conditions tab.

If the created Business Condition is currently satisfied, the Business Conditions tab displays “Current evaluation: true.” If the Business Condition is not satisfied, meaning the Rule is not triggered for response, the tab displays, “Current evaluation: fail.”



To remove a Business Condition:

- 1 Right-click on the Business Condition.

- 2 Click **Remove**.

If you do not see a **Remove** link, click the Delete Business Conditions **X** icon in the upper-right corner of the Business Conditions tab.

Confirm that you would like to delete the Business Condition.

- 3 Select **Save** from the **File** menu.

## Add Answer Action

A Managed Answer Rule may have multiple Answer Actions. For more information on Answer Actions, please see [Intelligent Search Language Tuning Guide](#).

## Add Search Component

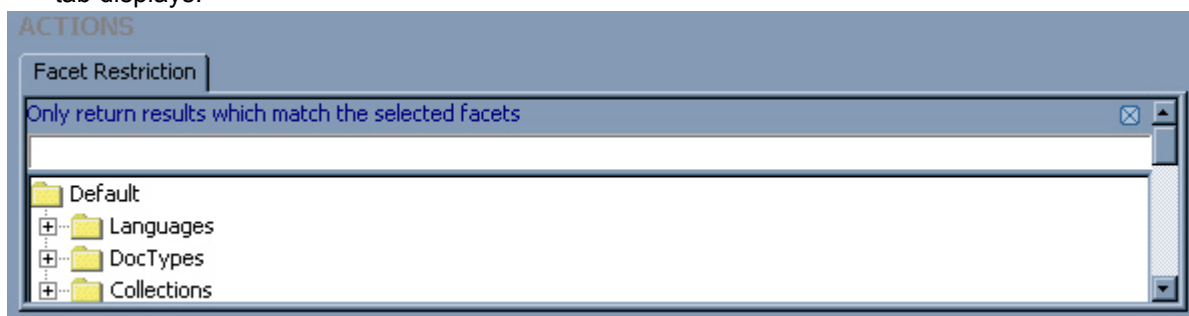
For more information on Search Component, please see [Intelligent Search Language Tuning Guide](#).

## Add Facet Restriction

You can restrict the scope of Search Actions by adding Facet Restrictions. Facets are created using the Personalized Navigation Setup. For more information, please see [Personalized Navigation Setup](#).

To add a Facet Restriction:

- 1 Select **Add Facet Restriction** from the **Advanced** menu of the Rule window. The Facet Restriction tab displays:



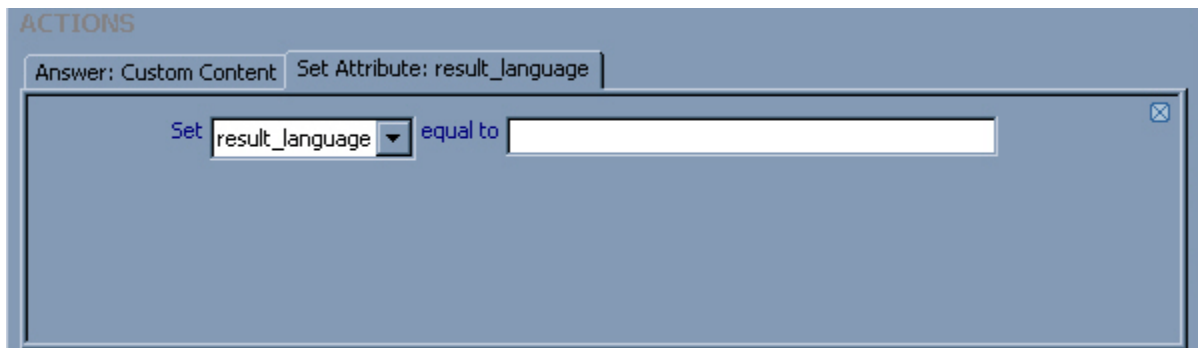
- 2 Select a facet from the tree structure. Using a comma to separate facets, you may select more than one facet.
- 3 Select **Save** from the **File** file menu of the Rules window.

## Add Set Attribute

The Set Attribute option allows you to use Business Condition context variables for use within subsequent Rules. In order to test Dictionary Objects that make use of Business Conditions using Test Drive, it is necessary to establish Rules with Set Attributes. Using a Rule with a Set Attribute action allows you to simulate the passing of a context variable through the User Interface. For more information on context variables, please see [Chapter A, Configuring Access to Business Data](#).

To add a Set Attribute:

- 1 Select **Add Set Attribute** from the **Advanced** menu of the Rule window. The Set Attribute tab displays:



- 2 Select a context variable from the drop down menu.
- 3 Enter the value of the context variable.
- 4 Select **Save** from the **File** file menu of the Rules window.'

## Add Relative Weights

You can specify custom values for weighting the score components used to rank unstructured search results by adding Relative Weights.

You can apply relative weights to the current Rule (local), or to all Rules that specify the Search method (global).

In order to specify local relative weights the Rule must contain an Answer Action using the **Search** method. For more information, please see [Search Answer Action Method on page 12](#).

To specify global relative weights create a Relative Weights action without an accompanying Search Action. For example, create a new Rule named "Relative Weights" with a question pattern and no action other than "Scope Weights". The Relative Weights parameters apply to all unstructured search results, unless they result from a Rule with specific Relative Weights settings.

You can specify Relative Weights based on any language conditions and available Business Conditions. For more information, please see [Add Business Conditions on page 5](#).

To add Relative Weights:

- 1 Select **Add Relative Weights** from the **Advanced** menu of the Rule window. The Relative Weights tab displays:

Field	Value	Percentage
Search Components	90	90.0%
Relevance of documents	0.000001	0.0%
Recency of documents	0	0.0%
Proximity of search terms	9.999999	10.0%
<b>Total</b>	<b>100</b>	<b>100.0%</b>

- 2 Enter a percentage to weight the Search Component score relative to the Relevance and Recency scores in the % **Search Components** field. Default is 100%.
- 3 In the **Relevance of documents** field, specify a percentage to weight the relevance score relative to the Search Component, Recency, and Proximity scores. Default is 0.0%.
- 4 In the **Recency of documents** field, specify a percentage to weight the Recency score relative to the Search Component, Relevance, and Proximity scores. The Recency score relies on the date/time stamp of a file. Default is 0%.
- 5 In the **Proximity of search terms** field, specify a percentage to weight the proximity of search terms score relative to the Search Component, Relevance and Recency scores. Proximity is only calculated for the configured number of results. returned. Default is 0.0%.

**Note:** The combination of all scores must equal 100%.

For more information on Search Component Scores and Document Relevance and Recency scores, please see [Chapter A, Configuring Access to Business Data](#).

## Add Scope Weights

You can specify custom values for weighting the document scope in which an answer is located using the Add Scope Weights option. The Rules Engine uses scope weights to calculate the total score for a response during scoring process.

You can apply scope weights to the current Rule (local), or to all Rules that specify the Search method (global).

You specify weights only for the current Rule by including a Scope Weights action and an Answer Action using the Search method in the same Rule. For more information, please see [Search Answer Action Method on page 12](#).

You specify global scope weights by specifying the Scope Weights action without an accompanying Search Action. The Scope Weights parameters applies to all unstructured search results, unless they result from a Rule with specific scope weight settings.

To add a Scope Weight:

- 1 Select **Add Scope Weights** from the **Advanced** menu of the Rule window. The Scope Weights tab displays:

**ACTIONS**

Answer: Search | **Scope Weights**

100.0% 0.0%

100 % Document

0 % Section

0 % Sentence

Total: 100 %

Remainder: 0 %

How important are these search factors relative to each other?

- 2 In the % **Document** field, specify a percentage to weight the score when an answer is located anywhere within the document.
- 3 Specify a percentage to weight the score when an answer is located within a section of the document in the % **Section** field.
- 4 Specify a percentage to weight the score when an answer is located within a sentence in a document in the % **Sentence** field.

**Note:** The combination of all scores must equal 100%.

## Add Suppress Rules

You can specify the suppression of subsequent Rules using the Suppress Rules option. The Suppress Rules tab provides options to:

- Suppress processing of all subsequent Rules
- Suppress processing for only the specified Rules

To suppress Rule processing:

- 1 Select **Add Suppress Rules** from the **Advanced** menu of the Rule window. The Suppress Rules tab displays:

The screenshot shows the 'BA Information Manager' window with the 'Advanced' menu open. The 'Suppress Rules' tab is selected. The window displays the following fields and sections:

- Name:** BA Information Manager
- Description:** (Empty text box)
- Notes:** Should return link when user queries install IM or Install information manager
- Enabled:** ☒ Enabled
- Domain:** Base Standard (Dropdown menu)
- CONDITIONS**
  - Question Patterns** (Selected tab)
  - Enter IML condition for matching questions: `<set up>, <information manager>`
  - # Matches:** Default (Dropdown menu)
- ACTIONS**
  - Answer:** Custom Content | Suppress Rules (Selected tab)
  - If this Rule matches the search query:
    - ☐ Stop looking for more Rules
  - Ignore the following Rules:
    - Add Rules by dragging items from a Search Results list, or by dragging the banner of a Rule window.
    - Remove items using the right-click menu.

- 2 Specify to terminate Rule matching or Rules to exclude from processing.
  - To terminate Rule matching, check the **Stop looking for more Rules** box.
  - To exclude a Rule from processing after this Rule has fired, drag and drop the Rule icon from within the Rule's window or from a Dictionary Find Tool result list, to the list at the bottom of the Suppress Rules tab.

You can delete Rules from the suppression list by right-clicking and selecting Remove from List, doing so prevents further answers from being displayed.

## Answer Action Methods for Rules

The following section explains Rules using the following Answer Action Methods:

- [Specific Excerpt Answer Action Method on page 12](#)
- [Search Answer Action Method on page 12](#)
- [Structured Query Answer Action Method on page 13](#)

- [Glossary Answer Action Method on page 14](#)
- [Plug-in Answer Action Method on page 15](#)
- [Process Wizard Answer Action Method on page 16](#)

For more information on creating Rules, please see [Intelligent Search Language Tuning Guide](#).

## Specific Excerpt Answer Action Method

The *Specific Excerpt* Answer Action is used to display specific excerpts within the Content Store.

To use the Specific Excerpt Answer Action:

- 1 Select **Specific Excerpt** from the drop down menu of the **Method** field of the Rule window. The Answer: Specific Excerpt tab displays:

- 2 Enter text in the **Custom Link Text** to display as the actual text of the link to the answer. For example to display the link text Oracle Knowledge Customer Support instead of [www.inquiria.com/support/index.htm](http://www.inquiria.com/support/index.htm).

**Note:** The Custom Link Text field is only valid when used with the Link Excerpt and Raw Excerpt algorithms. For more information please see [Oracle Knowledge Intelligent Search Administration Guide](#).

- 3 In the **Title** field, enter text to match within excerpt titles.
- 4 In the **Specific Excerpt** field, enter text to match within excerpts.
- 5 Select **Save** from the **File** menu in the Rule window.

## Search Answer Action Method

The *Search* Answer Action is used to search within indexed unstructured content.

An example of a Search Answer Action is the Launch Document Search Rule. The Launch Document Search Rule does not use any IML, but retrieves all unstructured search results using default ranking.

**Note:** If you create additional Rules with the Search Answer Action Method, you need to suppress the Launch Document Search Rule. Otherwise, you may see the same document retrieved multiple times by different Search Component Rules.

To use the Search Answer Action:

- 1 Select **Search** from the drop down menu of the **Method** field of the Rule window. The Answer: Search tab displays

The screenshot shows the 'ACTIONS' tab in a software interface. At the top, it says 'Answer: Search'. Below this, there are two dropdown menus: 'Purpose' set to 'Answer' and 'Method' set to 'Search'. To the right of the 'Method' dropdown is a small 'X' icon. Below these is a 'Priority' dropdown set to 'Highest'. Underneath is a section titled 'Retrieve document excerpts using this IML:' followed by a large empty text box. Below that is a checkbox labeled 'Retrieve document excerpts using default ranking', which is currently unchecked. At the bottom, there is a label 'Number of results must not exceed:' followed by a dropdown menu set to 'default'.

- 2 Enter valid IML to specify matching criteria for the excerpts generated by the Rules Engine in the **Retrieve document excerpts using this IML**.
- 3 (Optional.) Check the **Retrieve document excerpts using default ranking** to return results using the standard Search Component assignment and scoring process.

**Note:** You can use either the IML field or the Default Ranking option independently but, they *do not* work in conjunction with one another.

- 4 Select a value from the **Number of results must not exceed** drop down menu.
- 5 Select **Save** from the **File** menu in the Rule window.

## Structured Query Answer Action Method

The *Structured Query* Answer Action is used to specify retrieval from configured sources of structured data.

To use the Structured Query Answer Action:

- 1 Select **Structured Query** from the drop down menu of the **Method** field of the Rule window. The Answer: Structured Query tab displays

The screenshot shows the 'ACTIONS' window for the 'Answer: Structured Query' method. The window contains the following fields and options:

- Purpose:** Answer (dropdown)
- Method:** Structured Query (dropdown)
- Priority:** Highest (dropdown)
- Data Source:** (dropdown)
- Title of results:** (text field)
- Description:** (text field)
- Structured Query:** (text field)
- IML to restrict Structured Query:** (text field)
- Retrieve structured results using default ranking:** ☐
- Table rows to display with description:** 3 (text field)
- Number of results must not exceed:** default (dropdown)

- 2 Select a **Data Source** to query from the drop down menu. Please see the [Oracle Knowledge Intelligent Search Administration Guide](#) for information on configuring data sources for structured information retrieval.
- 3 In the **Title of results** field, enter text to specify a title under which the query results are displayed.
- 4 In the **Description** field, enter text to specify an optional description for the query results.
- 5 In the **Structured Query** field, enter the SQL query to submit.
- 6 (Optional.) Enter IML in the **IML to restrict Structured Query** to restrict the results of the query. The Rules Engine filters the query results based on matching the specified IML.
- 7 (Optional.) Check the **Retrieve structured results using default ranking** if you want to return results using the standard Search Component assignment and scoring process.
- 8 Enter the number of **Table rows to display with description**.
- 9 Select a value from the **Number of results must not exceed** drop down menu.
- 10 Select **Save** from the **File** menu in the Rule window.

## Glossary Answer Action Method

The *Glossary* Answer Action is used to display terms and definitions from prepared glossaries. For more information, please see [Alias Lists on page 16](#).

**Note:** Oracle Knowledge provides a preconfigured Rule, "Definition of" to use with the Alias List called "Glossary."



To use the Glossary Answer Action:

- 1 Select **Glossary** from the drop down menu of the **Method** field of the Rule window. The Answer: Glossary tab displays:

- 2 Select the alias list from the **Glossary Alias List to use** drop down menu.
- 3 In the **Retrieve a glossary entry for the term referred to by this variable** enter the variable set in the Rule condition that corresponds to the desired glossary entry.
- 4 Select **Save** from the **File** menu in the Rule window.

## Plug-in Answer Action Method

The *Plug-in* Answer Action is used to create a response using an external method.

To use the Plug-in Answer Action:

- 1 Select **Plug-in** from the drop down menu of the **Method** field of the Rule window. The Answer: Plug-in tab displays:

- 2 Select a **Plugin** from the drop down menu.
- 3 If displayed, enter text in the **Question**, **type**, or **File** field.  
The **Description** field provides an explanation of the plug-in functionality. If displayed, enter text in the field below **Description**, labelled **Question**, **type**, or **File**, depending on the plug-in. For installed examples, search for all Rules whose Methods include `Plug-in`.

- 4 Select **Save** from the **File** menu in the Rule window.

## Process Wizard Answer Action Method

The *Process Wizard* Answer Action is used to display a Process Wizard in response to a query.

To use the Process Wizard Action:

- 1 Select **Process Wizard** from the drop down menu of the **Method** field of the Rule window. The Answer: Process Wizard tab displays:

- 2 Select the process wizard you want to use from the drop down menu.
- 3 Select **Save** from the **File** menu in the Rule window.

## Alias Lists

*Alias Lists* are Dictionary Objects that provide a means of substituting one term for another. Alias Lists are transformative where, unlike Concepts, a word or abbreviation in a user query is replaced with an associated value. For example, the existence of the delivered US State Abbreviations Alias List associates a user query containing "NY" with "New York."

Creating Alias Lists in Dictionary Manager allows Oracle Knowledge Intelligent Search to:

- Provide definitions to key terms using the [Answer Action Methods for Rules on page 11](#).
- Identify values for common abbreviations such as, state names or stock symbols.


**Note:** If you use multiple alias lists to launch glossaries that are restricted to different audiences, the same term cannot appear in multiple glossaries. Including the same terms in different alias lists results in none of the alias lists launching on Information Manager.

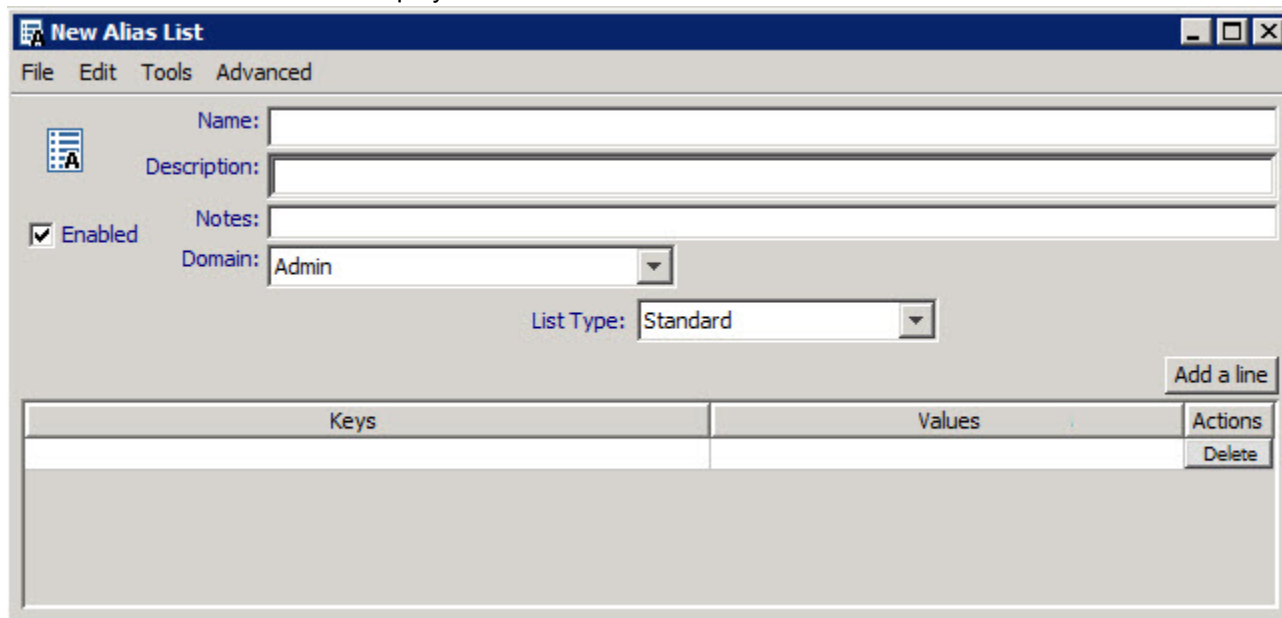
## Create a New Alias List

By creating new Alias Lists, you allow Oracle Knowledge Intelligent Search to recognize terms or abbreviations as having a set value.

Oracle Knowledge provides the VIL Alias List function. For more information, see [ALIAS The Alias Function on page 32](#) of the *Intelligent Search Language Reference Guide* and [Create a Rule to Fire When an Alias List Key is Matched](#) and [Pass User-Selected Facets Through an Alias List on page 18](#) of this guide.

To create a new Alias List:

- 1 In Dictionary Manager, click the New Alias List icon  or select **New Alias List** from the File menu. The Alias List window displays:



- 2 Enter a descriptive name for the Alias List in the **Name** field.
- 3 (Optional) Fill in the **Description** field.
- 4 The Description field has no impact on Oracle Knowledge Intelligent Search results. It is used to clarify the intended function of the Alias List among users.
- 5 (Optional) Enter any additional information pertinent to the Alias List in the **Notes** field.
- 6 The Notes field has no impact on Oracle Knowledge Intelligent Search results. It is used to track comments on Rule modification among users.
- 7 Check the **Enabled** box. (This is checked by default.) If the **Enabled** box is not checked the Alias List is ignored by Oracle Knowledge Intelligent Search.
- 8 Select the **Domain** from the drop down list. Only Domains you have permission for are displayed. The Domain field is set to the most recently saved Domain by default. For more information on Domains see the preceding section in [Intelligent Search Language Tuning Guide](#).
- 9 Select a **List Type** from the drop down menu. Choose from standard, Root Concept Property, or Context Variable Property.
  - Standard is the default.
  - Root Concept Alias Lists create a list that contains ONLY children of a root concept defined in the Root Concept field. For example: Product and Product types.
  - Context Variable Alias Lists create a list that corresponds to the context variable defined from the drop down menu in the Context Variable field. For example: User Location and State Abbreviations.
- 10 Enter a word, phrase, Concept, or global variable in the **Keys** column.

**Note:** To reference a Concept you must use angle brackets, for example, <information manager>. To reference a Global Variable you must use the pound sign, for example #GLOBALVARIABLE.

- 11 Enter a value in the value column for Oracle Knowledge Intelligent Search to associate with the corresponding key term.

## Create a Rule to Fire When an Alias List Key is Matched

To create a rule that fires only when an alias list key is matched, set up a question pattern that matches every key in the alias list. Due to the potential size an alias list could be, doing this manually will be hard to maintain. Instead, use a VIL expression in the question pattern box to accomplish this automatically:

```
{"dummyString"->ALIAS (LIST="aliasListName", KEYOUTONLY="TRUE") ->STRING (SEP=" OR " ) }
```

This expression populates the question pattern with all the alias list keys separated by the OR operator. The "dummyString" can be any string; it is required for the VIL expression to work but plays no part in the eventual IML pattern. The "aliasListName" is the name of whatever alias list contains the desired keys.

This makes maintenance easier now because any additions, deletions, or modifications to the alias list's entries are automatically updated in the question pattern.

For more information, see [ALIAS The Alias Function on page 32](#) of the Intelligent Search Language Reference Guide.

## Pass User-Selected Facets Through an Alias List

Use the VIL function FACET() to get a string of all the Facets, or a subset of them, that a user has selected and then pass the output to the alias list function, e.g. {"A->FACET()->ALIAS(LIST="someList")}

**Note:** The output of FACET() cannot be used for concept matching, so the keys in the Alias List must be strings.

For more information, see [ALIAS The Alias Function on page 32](#) of the Intelligent Search Language Reference Guide.

## Alias List Tools

Before saving an Alias List, you can verify the function of the Alias List by selecting [Validate Alias Lists on page 19](#) from the **Tool** menu in the **Alias List** window.

You may also choose to [Import Text Files on page 18](#).

## Import Text Files

To import existing Alias Lists to be used by your current version of Oracle Knowledge Intelligent Search:

- 1 From Dictionary Manager, select **Import from file** from the **Tools** menu of the Alias Lists window. The Import Alias Lists file finder displays.
- 2 Navigate to the directory containing the .xml file containing Alias Lists and select the file.
- 3 Select **Import**.  
The imported Alias List displays.

**Note:** Typically, the Import Alias List functionality is used to re-import Alias Lists exported as XML files.

## Validate Alias Lists

After selecting **Validate Alias List** from the **Tools** menu, Oracle Knowledge Intelligent Search automatically checks the validity of your Alias List. If the Alias List is valid the following **Alias List is Valid** dialog displays.



## Advanced Feature of Alias Lists

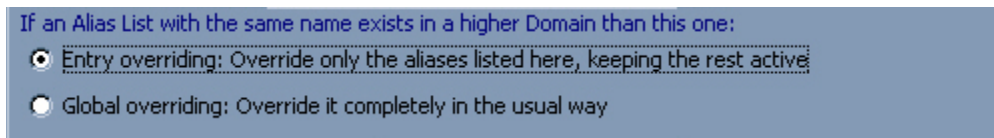
This section explains how to use the Alias Lists advanced feature, [Set Overriding](#).

### Set Overriding

Alias Lists with the same name and values can belong to multiple Dictionary Domains. If Alias Lists with the same name occur, the default override behavior is determined by the order of Domains in the Domain List. For more information on Domains and Domain Lists, please see [Intelligent Search Language Tuning Guide](#).

Using the Set Overriding option, you can specify custom override behavior.

- Select **Set Overriding** from the **Advanced** menu of the Alias Window. The overriding options append to the Alias List window:



- Choose between Entry Overriding or Global Overriding.

Entry Overriding specifies that Keys and Values assigned in the current Alias List are override entries with the same Keys and Values of Alias Lists of the same name in higher ranked Domains.

Global Overriding specifies that the current Alias List overrides an Alias List with the same name in a higher ranked Domain for all entries, not just for the Keys and Values the two Alias Lists share.

## Exception Lists

Use *Exception Lists* to create exceptions to the stemming process used by Oracle Knowledge Intelligent Search. The stem of a word is the root form of the word—the word minus any inflection. For example, "work" is the stem of "works," "working," and "worked." Stemming accounts for changes in word shape owing to the context in which words occur without altering the meaning of the word.

In English, stemming covers the relations represented in the following table—for noun, plural; for verbs, person and tense inflection; for adjectives, comparatives:

Part of Speech	Stem	Inflected Form
noun	dog	dogs
verb	carry	carried

Part of Speech	Stem	Inflected Form
verb	carry	carries
verb	carry	carrying
adjective	nice	nicer

Stemming does not cover the relation between the words in columns **A** and **B** in the following table. Unlike the previous list, these are meaning-altering relations: the words in **B** add some meaning component to those in **A**.

A	B
love	lover
love	lovable
Russia	Russian
italic	italicize
romantic	romanticism
stupid	stupidity
fool	foolish

The LLLA (Low Level Language Analysis) Service of Oracle Knowledge handles stemming.<sup>1</sup> You can see the stem of a word in Test Drive, Annotations, Stem.

In some cases, it is necessary to add a stemming exception or modify the basic stemming rules. A typical case arises with acronyms. For example, the Product name “Ikes” is stemmed to “Ik”, as Oracle Knowledge Intelligent Search recognizes -es as a suffix. To prevent this we must add “Ikes” to the Exception List as described in the following section.

1. Third party software handles stemming for foreign languages.

## Edit an Exception List

To edit an Exception List:

- 1 Using the Dictionary Find Tool, select the language specific Exception List. The Exception List displays:

Stems	Tokens	Actions
aardwolf	aardwolves	Delete
abacus	abaci, abacuses	Delete
abash	abashed, abashing	Delete
abducens	abducens	Delete
abecedarian	abecedarian	Delete
abet	abets, abetted, abetting	Delete
abhor	abhorred, abhorring, abhors	Delete
abide	abided, abides, abiding, abode	Delete

**Note:** If the language you want to edit does not appear, please see [Create a New Exception List on page 22](#).

- 2 Click **Add a line**. A new line appends to the bottom of the list.
- 3 In the **Stems** column, enter the preferred stemmed form of the word.
- 4 In the **Tokens** column, enter the word as it appears in a user query. Using a comma to separate the words, you may enter more than one form per stem. A token consisting of multiple parts may be added to the Exception List if the associated stem consists of the same number of words.

Stems	Tokens	Actions
ikes	ike,ikes	Delete

**Note:** Exception Lists are case sensitive. A token entered using capitals matches only strings with the same capitalization; however, if the token is entered in lower case it matches a string in any case.

- 5 Select **Save**, from the File menu of the Exception List window. The newly created exception appears alphabetically.

**Note:** A stem may only appear once on the Exception List. Upon saving, a stem listed twice gets condensed to one instance with the associated tokens appearing on one line separated by a comma.

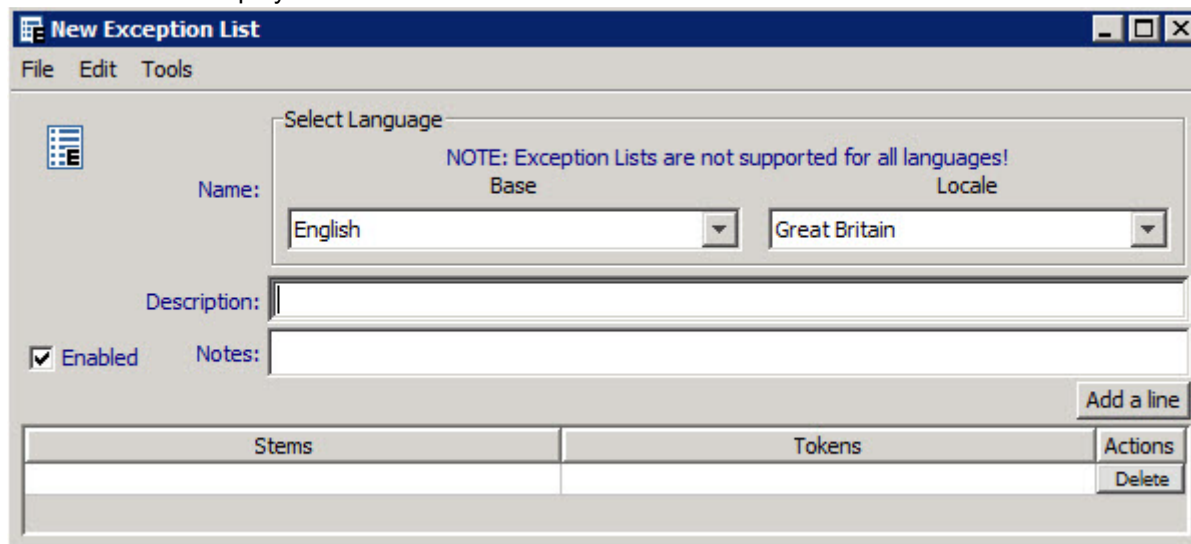
**Important!** Modifications to an Exception List do not affect live Intelligent Search results until you [Commit Dictionary Changes to Central Repository \(vroot\)](#), perform a full content re-index, and synch the changes to the production environment. To see the effects of a new or modified Exception List without committing, use [Test Drive](#). For more information, see [Oracle Knowledge Intelligent Search Administration Guide](#).

## Create a New Exception List

**Note:** Exception Lists can be created at both a *base* language level, such as English, French, Japanese, or at a *locale* level, such as English US, English GB, French Canada, French France. If the language of content being indexed is specified as a base language, then only the base language Exception List is used. If the language is specified as a locale language, then the locale Exception list, if one exists, is merged with the base language Exception List. Where any conflicts arise, the locale values are used.

To create a new Exception List:

- 1 In Dictionary Manager, select the  icon or **New Exception List** from the **File** menu. The Exception List window displays:



- 2 Select the Language Base and Locale the Exception List is intended for from the drop down menus in the **Name** field.

Exception Lists can be created at both a Base language level, such as English, French, Japanese, etc. or at a Locale level, such as English US, English GB, French Canada, French France, etc. If the language of content being indexed is specified as a Base language, then only the Base language Exception List is used; if the language is specified as a Locale language, then the Locale Exception list, if it exists, is merged with the Base language Exception List. Where any conflicts arise, the Locale values are used.

**Note:** If the language you desire does not display in the drop down menu, then an Exception List for the language has been created. Please see [Edit an Exception List on page 21](#).

- 3 (Optional.) Enter a definition into the **Description** field.  
The **Description** field is used to clarify the meaning among users, and can be used to find dictionary objects.
- 4 (Optional). Enter any information regarding the Exception List in the **Notes** field.
- 5 The **Notes** field has no impact on Oracle Knowledge Intelligent Search results. It is used to track comments on dictionary object modification among users.
- 6 Check the **Enabled** box. (This is checked by default.) If the Enabled box is not checked, the Exception list is ignored by Oracle Knowledge Intelligent Search.
- 7 In the **Stems** column, enter the preferred stemmed form of the word.



- 8 In the **Tokens** column, enter the word as it appears in a user query. Using a comma to separate the words, you may enter more than one form per stem.

Stems	Tokens	Actions
ikes	ike,ikes	Delete

- 9 To add exceptions, select **Add a line**. A new line appends to the bottom of the list.
- 10 Select **Save**, from the File menu of the Exception List window. The newly created exception appears alphabetically.

**Important!** Modifications to an Exception List do not affect live Intelligent Search results until you perform, a full content reindex, and the changes have been synchronized to the production environment. For more information, see [Oracle Knowledge Intelligent Search Administration Guide](#). To see the effects of a new or modified Exception List without committing, use [Test Drive](#).

## Exception List Tools

To import an existing Exception List:

- 1 Select **Import from file** from the **Tools** menu in the Exception List window.  
The **Import Exception List** file finder displays.
- 2 Select the Exception List as a \*.csv file, and select **Import**. The Exception List imports.<sup>1</sup>  
**Import from file** is usually employed to re-import previously exported Exception Lists.  
Use **Export to file** to export exception lists for back up purposes or to import into another instance.

## Spell Check Lists

Use *Spell Check Lists* to provide user-defined spelling corrections to be applied by the spell-checker in Oracle Knowledge Intelligent Search. Spelling corrections entered in the Spell Check List are added to the corrections provided by the installed spell-checker.

1. Third party software handles stemming for foreign languages

## Edit a Spell Check List

To edit a Spell Check List:

- 1 Using the Dictionary Find Tool, select the language specific Spell Check List. The Spell Check List displays:

Misspelling	Correction	Case Sens	Actions
denormaliez	denormalize	<input checked="" type="checkbox"/>	Delete
denormalize	denormalize	<input checked="" type="checkbox"/>	Delete
denormalized	denormalized	<input checked="" type="checkbox"/>	Delete
whats	what's	<input checked="" type="checkbox"/>	Delete

Note: all values will be normalized (accents removed, etc.) before save

**Note:** If no Spell Check List exists for the language, you can create a new Spell Check List from the File menu, by clicking “New Spell Check List” and selecting the desired Language and possibly, Locale. Please see [Create a New Spell Check List on page 25](#).

- 2 Click **Add a line**. A new line appends to the bottom of the list.
- 3 In the **Misspellings** column, enter the misspelling that you want to correct.
- 4 In the **Correction** column, enter the corrected form of the word.
- 5 If the “Match Case” column is checked, the correction will be applied to match the case of the misspelling.
- 6 Select **Save** from the File menu of the Spell Check List window.

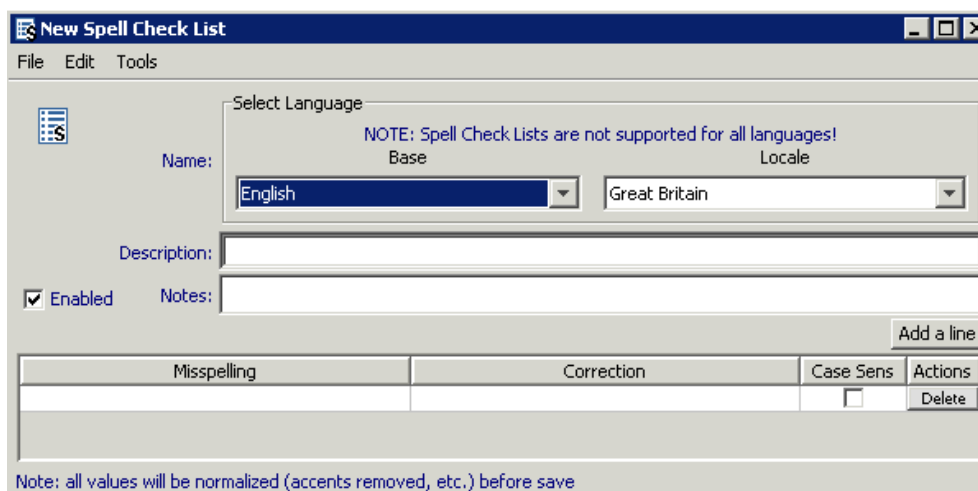
**Note:** You may only correct a single word. You cannot enter a phrase in either the **Misspelling** nor the **Correction** columns.

## Create a New Spell Check List

**Note:** Spell Check Lists are language specific. They can be created at both a 'base' language level, such as English, French, Japanese, etc. or at a 'locale' level, such as English US, English GB, French Canada, French France, etc. If the language of content being indexed is specified as a 'base' language, then only the base language Spell Check List is used; if the language is specified as a 'locale' language, then the 'locale' Spell Check list, if it exists, is merged with the 'base' language Spell-Check List. Where any conflicts arise, the 'locale' values are used.

To create a new Spell Check List:

- 1 In Dictionary Manager, select the  icon or **New Spell Check List** from the File menu. The Spell Check List window displays:



- 2 Select the language the Spell Check List is intended for from the drop down menus.

**Note:** If the language or locale does not display in the drop down menu, then a Spell Check List for the language or locale has been created, or the language or locale has not been enabled. See [Edit a Spell Check List on page 24](#).

- 3 (Optional) Enter a definition in the **Description** field. The **Description** field is used to clarify the meaning among users, and can be used to find dictionary objects.
- 4 (Optional) Enter any information regarding the Spell Check List in the **Notes** field.
- 5 The **Notes** field has no impact on Oracle Knowledge Intelligent Search results. It is used to track comments on dictionary object modification among users.
- 6 Check the **Enabled** box. This is checked by default. If the Enabled box is not checked, the Spell Check List is ignored by Oracle Knowledge Intelligent Search.
- 7 In the **Misspellings** column, enter the misspelling that you want to correct.
- 8 In the **Correction** column, enter the corrected form of the word.
- 9 If the "Match Case" column is checked, the correction will be applied to match the case of the misspelling.
- 10 Select **Save** from the File menu of the Spell Check List window.

---

## Spell Check List Tools

To import an existing Spell Check List:

- 1 Select **Import from file** from the **Tools** menu in the Spell Check List window.
- 2 Select the Spell Check List as a \*.csv file, and select **Import**. The Spell Check List imports.

To export the Spell Check List:

- 1 Select the lines in the Spell Check List that you wish to export. You can use `Ctrl-A` to select all the lines in the Spell Check List.
- 2 Select **Export to file** from the **Tools** menu in the Spell Check List window.
- 3 Select the folder, and enter the name of the .csv file that you want the Spell Check List to be exported to.
- 4 Click **Export**. The selected Spell Check List entries will be exported to the file.

## Intent Hierarchy

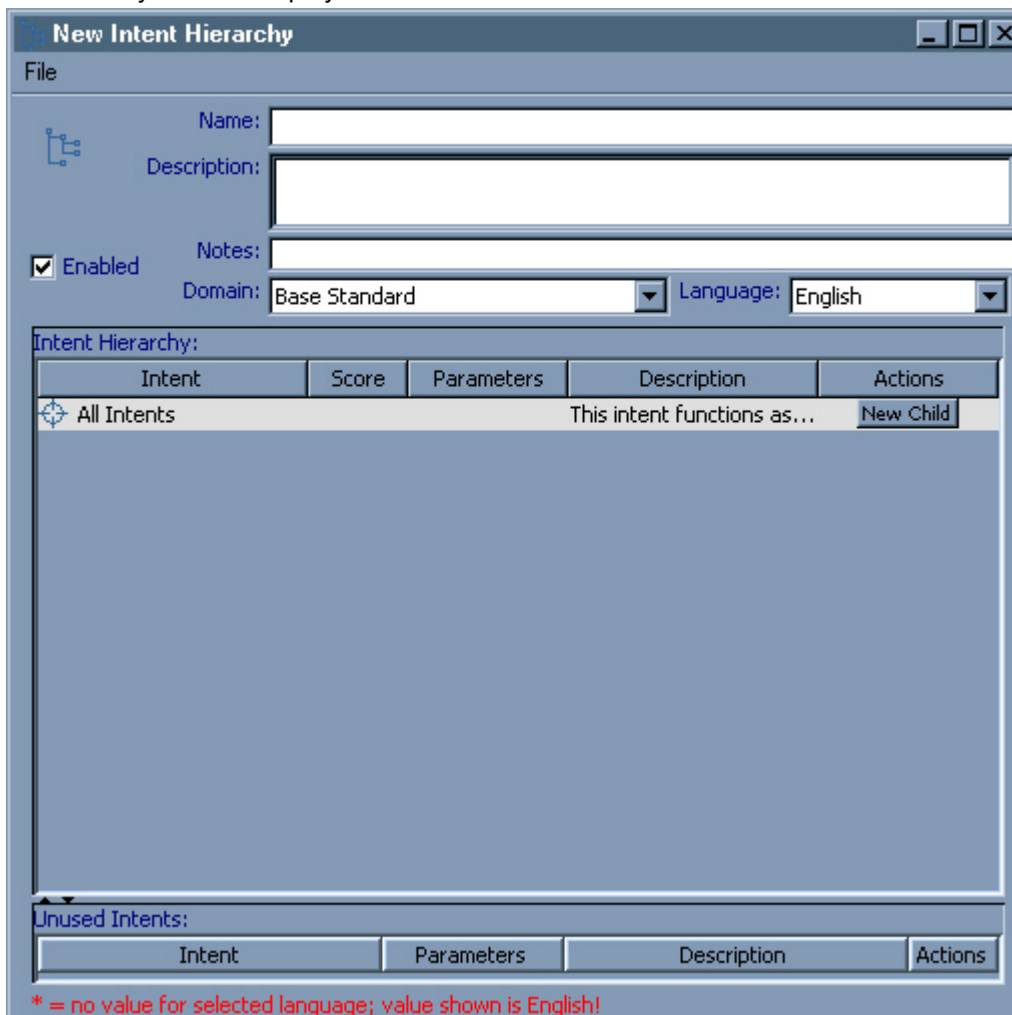
Intents are organized as parent, siblings, and child nodes in an Intent Hierarchy. Intent Hierarchies descend from a single node at the highest level called *All Intents*. The position of the Intent within the Intent Hierarchy, plus the score of the closest parent, determine which intent has precedence in firing a configured Intent Response. The Rules Engine selects the lowest-level matching Intent in the hierarchy as the best possible response. If a low-level Intent does not satisfy a user query, the Rule Engine works backwards to match on a higher-level Intent. This process, referred to as *degradation*, ensures that Oracle Knowledge Intelligent Search returns the most specific Intent Response to a user query. For more information on Intents, Intent Responses, please see [Intelligent Search Language Tuning Guide](#).

## Create a new Intent Hierarchy

Oracle Knowledge provides pre-configured Intent Hierarchies for various industries. These can be displayed using the **Find** functionality.

To create a new Intent Hierarchy:

- 1 In Dictionary Manager, select the  icon or **New Intent Hierarchy** from the File menu. The New Intent Hierarchy window displays:



**New Intent Hierarchy**

File

Name:

Description:

Notes:

☒ Enabled

Domain:  Language:

Intent Hierarchy:

Intent	Score	Parameters	Description	Actions
All Intents			This intent functions as...	New Child

Unused Intents:

Intent	Parameters	Description	Actions
--------	------------	-------------	---------

\* = no value for selected language; value shown is English!

- 2 Enter a descriptive name for the Intent Hierarchy in the **Name** field.
- 3 (Optional.) Fill in the **Description** field.
- 4 The Description field has no impact on Oracle Knowledge Intelligent Search results. It is used to clarify the intended function of the Intent Hierarchy among users.
- 5 (Optional.) Enter any information regarding the Intent Hierarchy in the **Notes** field.
- 6 The Notes field has no impact on Oracle Knowledge Intelligent Search results. It is used to track comments on Intent Hierarchy modification among users.
- 7 Check the **Enabled** box. (This is checked by default.) If the Enabled box is not checked the Intent Hierarchy is ignored by Oracle Knowledge Intelligent Search.

**Note:** When displaying an Intent Hierarchy, all of the intent names will display in red, with the message `*intent not found; name = xxxx`, if the Domain of the Intent Hierarchy is not included in the Active Domain List, displayed in the upper-right corner of the Dictionary Manager. To fix this, reset the Active Domain List; then close and reopen the Intent Hierarchy.

If a language-specific name for an intent has not been specified for the Language value of the Intent Hierarchy, the name is displayed in red as `*null` and its parameters will be displayed as `* <no name>`. Fix this by editing the Intent and adding names for the language.

- 8 Select the **Domain** from the drop down list. Only Domains you have permission for are displayed. The Domain field is set to the most recently saved Domain by default. For more information on Domains see the preceding section [Intelligent Search Language Tuning Guide](#).
- 9 Select the **Language** from the drop down menu.  
Additionally, you may specify a locale for the language.
- 10 Select **New Child** to add Intents to the new Intent Hierarchy. For more information on Intents, please see [Intelligent Search Language Tuning Guide](#).
- 11 Select **Save**, from the File menu of the User Profile window.

## User Profiles

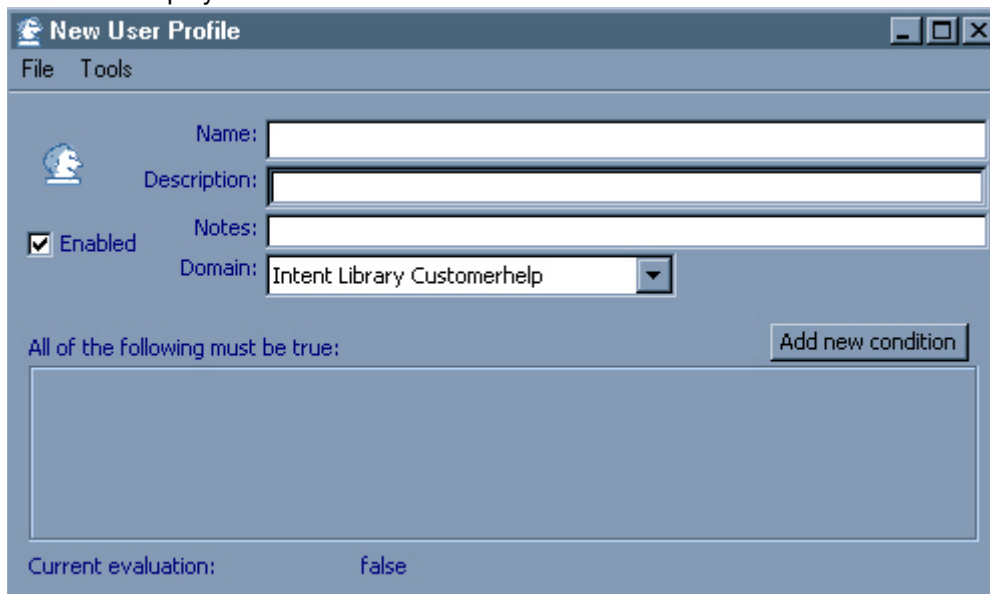
User Profiles are a type of Business Condition used to restrict an Intent Response. When using User Profiles, the set condition(s) must be satisfied by the user query in order to display the associated Intent Response.

For more information on Business Conditions, please see [Add Business Conditions on page 5](#). For more information on Intent Responses, please see [Intelligent Search Language Tuning Guide](#).

## Create a new User Profile

To create a new User Profile:

- 1 In Dictionary Manager, select the  icon or **New User Profile** from the File menu. The New User Profile window displays:



- 2 Enter a descriptive name for the User Profile in the **Name** field.
- 3 (Optional.) Fill in the **Description** field.
- 4 The Description field has no impact on Oracle Knowledge Intelligent Search results. It is used to clarify the intended function of the User Profile among users.
- 5 (Optional.) Enter any information regarding the User Profile in the **Notes** field.
- 6 The Notes field has no impact on Oracle Knowledge Intelligent Search results. It is used to track comments on User Profile modification among users.
- 7 Check the **Enabled** box. (This is checked by default.) If the Enabled box is not checked the User Profile is ignored by Oracle Knowledge Intelligent Search.
- 8 Select the **Domain** from the drop down list. Only Domains you have permission for are displayed. The Domain field is set to the most recently saved Domain by default. For more information on Domains see the preceding section [Intelligent Search Language Tuning Guide](#).
- 9 Click **Add new condition**. The Business Condition window displays.
- 10 Complete the Business Conditions as described in [Add Business Conditions on page 6](#).
- 11 Select **Save**, from the File menu of the User Profile window.

## User Profile Tools

Before saving a User Profile, you can verify the function of the User Profile by selecting an [Validate Profile on page 27](#) from the **Tool** menu in the **User Profile** window.

## Validate Profile

After selecting **Validate Profile** from the **Tools** menu, Oracle Knowledge Intelligent Search automatically checks the validity of your User Profile.

## Show Concepts Tree

The Concepts tree displays hierarchies of Concept type of or part of relationships defined in the Dictionary. The hierarchies originate at specified Root Concepts, as described in [Specifying the Root Concepts for the Dictionary Manager](#). The hierarchy displayed below each Root Concept reflects the Concept relationships defined below it in the Ontology.

You can edit concepts by selecting them directly from the Concepts tree, and you can add concepts to other Dictionary objects by dragging and dropping them directly from the Concepts tree.

## Subject Maps

Subject Maps are used in conjunction with the language and subject parameters passed with a user query to determine which Domain List to use in processing the query. Subject Maps are used to associate the pairings



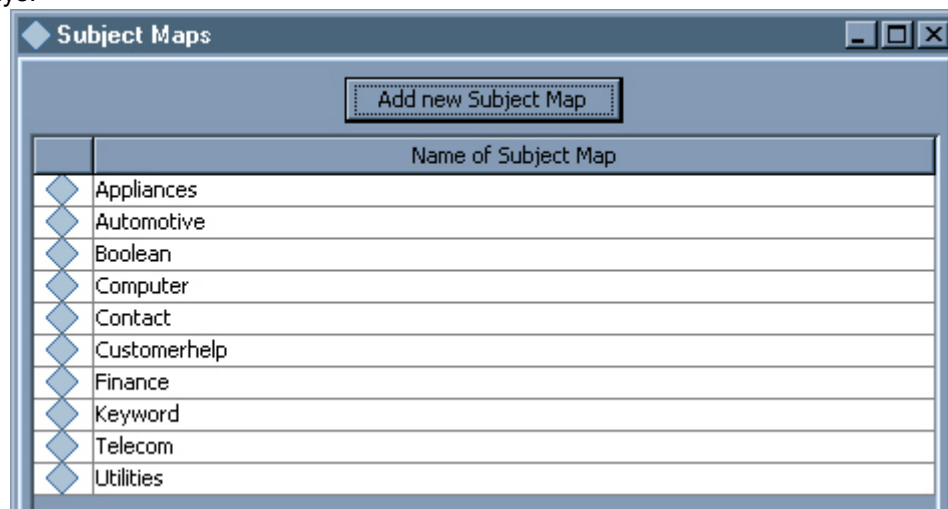
of a subject area and language with a Domain Lists. For example, the Automotive Subject Map contains the automotive Domain Lists for each language.

Language	Domain List	Actions
<b>Default</b>	English Automotive	Delete
<b>Chinese China</b>	Chinese Simplified Automotive China	Delete
<b>Chinese Simplified</b>	Chinese Simplified Automotive	Delete
<b>Chinese Taiwan</b>	Chinese Traditional Automotive Taiwan	Delete
<b>Chinese Traditional</b>	Chinese Traditional Automotive	Delete
<b>English</b>	English Automotive	Delete
<b>English Great Britain</b>	English Appliances Great Britain	Delete
<b>English United States</b>	English Appliances United States	Delete
<b>French</b>	French Automotive	Delete
<b>French France</b>	French Automotive France	Delete
<b>German</b>	German Automotive	Delete
<b>German Germany</b>	German Automotive Germany	Delete
<b>Italian</b>	Italian Automotive	Delete
<b>Italian Italy</b>	Italian Automotive Italy	Delete
<b>Japanese</b>	Japanese Automotive	Delete
<b>Japanese Japan</b>	Japanese Automotive Japan	Delete
<b>Korean</b>	Korean Automotive	Delete
<b>Korean Korea</b>	Korean Automotive Korea	Delete
<b>Portuguese</b>	Portuguese Automotive	Delete
<b>Portuguese Brazil</b>	Portuguese Automotive Brazil	Delete

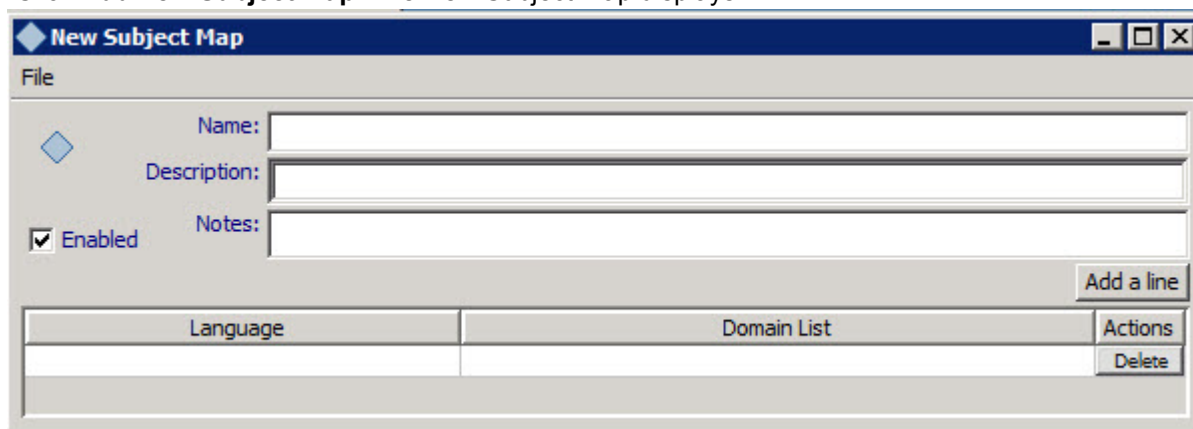
## Edit Subject Maps

To edit Subject Maps:

- 1 In Dictionary Manager, select **Edit Subject Maps** from the **Tools** menu. The Subject Maps window displays:



- 2 Click **Add new Subject Map**. The New Subject Map displays:



- 3 Enter a descriptive name in the **Name** field.
- 4 (Optional.) Enter a definition into the **Description** field.
- 5 The **Description** field is used to clarify the meaning among users, and can be used to find dictionary objects.
- 6 (Optional). Enter any information regarding the Subject Map in the **Notes** field.  
The **Notes** field has no impact on Oracle Knowledge Intelligent Search results. It is used to track comments on Concept modification among users.
- 7 Check the **Enabled** box. (This is checked by default.) If the Enabled box is not checked, the Subject Maps is ignored by Oracle Knowledge Intelligent Search.
- 8 Select a **Language** from the drop down menu.

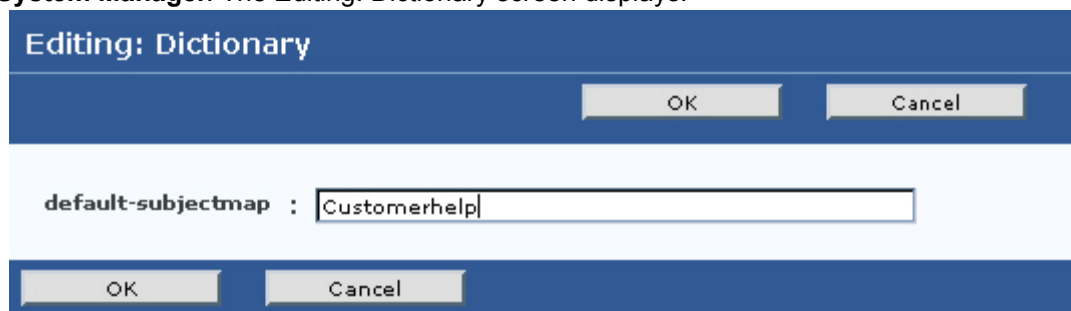
**Note:** Subjects Map can contain only one Domain List per language.

- 9 Select a **Domain List** from the drop down menu.
- 10 Click **Add a line**, to add Domain Lists for other languages.

## Change the Default Subject Map

The default Subject Map is used to determine which Domain List is used to process user requests, when the Subject is not specified for a request. To set the default Subject Map:

- 1 Select **Default subject map** from the **Instance** section of the **Advanced Configuration Facility** in **System Manager**. The Editing: Dictionary screen displays:



- 2 Enter the name of the Subject Map created in Dictionary Manager in the **default-subjectmap** field.
- 3 Click **OK**. The changes are saved upon closing.
- 4 Restart Oracle Knowledge.

For more information on the Advanced Configuration Facility, please see [Oracle Knowledge Intelligent Search Administration Guide](#).

## Oracle Knowledge Language Analysis

Oracle Knowledge performs language analysis during both content processing and request processing. Content processing is the means by which Oracle Knowledge collects, standardizes, and indexes the application content. Request processing is the means by which Oracle Knowledge Intelligent Search processes and responds to user requests.

During content processing, Intelligent Search identifies the concepts that occur in the application content, performs additional semantic analysis, and records the resulting information as a set of indexes.

During request processing, Intelligent Search determines the context, conditions, and concepts that occur in user requests by matching the content of the request against the information in the Dictionary. It uses that information to construct a set of criteria that defines the best possible responses to the request. It then matches the best answer criteria to the content indexes, and retrieves the corresponding documents from the Content Store for display to the end user.

## Language Analysis Components

The Oracle Knowledge components that interact to perform language analysis include:

- The Content Store, which stores the application content and associated data
- The Language Analyzer, which performs basic semantic analysis of both user requests and application content

- The Dictionary, which contains the Concepts and Rules that set the criteria for the best possible answer
- The Rules Engine, which matches user requests to Rules by comparing request content to business conditions and language conditions, and generates a list of actions that the application must perform to generate the best possible response.
- The Evaluator, which scores and ranks the results of the various actions

[The Language Analysis Process on page 31](#) describes the interaction of the language analysis components during request processing.

## The Language Analysis Process

Oracle Knowledge performs language analysis during both content processing and request processing.

During content processing, Oracle Knowledge collects, standardizes, and indexes the application content. The Language Analyzer analyzes each word in the content and establishes its relationship to:

- The Concept information defined in the Dictionary
- The structure of the document in which it is located

The Language Analyzer stores the resulting semantic information in the Content Store for use during request processing.

During request processing, the Language Analyzer:

- Analyzes the request using the Concepts stored in the Dictionary

and the Rules Engine:

- Evaluates the request using the Rules stored in the Dictionary
- Compiles the list of actions, which are potential application responses to the request

The Evaluator:

- Evaluates the actions, and ranks the responses based on their scores, as described in [The Scoring Process on page 33](#)
- Formats the responses for display by the User Interface

## The Content Store and Indexes

During content processing, the Language Analyzer analyzes each word in the unstructured application content. The Language Analyzer stores the results of its analysis as a set of indexes.

## Analyzing Requests

When a user submits a question or request using Intelligent Search, the User Interface passes the request to the Rules Engine. The Rules Engine creates a sentence object that contains the user input and additional user profile and session information.

During sentence object creation, the Rules Engine calls the Language Analyzer. The Language Analyzer identifies the basic linguistic and semantic components of the user request. This information is used by the Rules Engine during request evaluation.

The Language Analyzer also resolves the meaning of pronouns and other implicit references to concepts that occurred in previous requests within the session, as described in [Evaluating Anaphora on page 35](#).

## Evaluating Requests

The Rules Engine evaluates the sentence object against the set of Rules contained in the Dictionary. The Rules specify language and business conditions that potentially match various aspects of user requests as described in:

- Language Conditions
- Business Conditions

The evaluation process includes the phases described in:

- [Evaluating Spelling on page 32](#)
- [Evaluating Business Conditions on page 32](#)
- [Evaluating Language Conditions on page 32](#)

### EVALUATING SPELLING

The Rules Engine evaluates the spelling of the words in the sentence object using a packaged spelling checker program. For more information on the spelling checker program, see the [Oracle Knowledge Intelligent Search Administration Guide](#).

### EVALUATING BUSINESS CONDITIONS

Business Conditions are special conditions within Rules that match contextual information about the user or the current session. During request processing, a component called the Preference Service collects contextual user and session information from configured sources, and adds this information to the sentence object.

**Note:** You must have the Business Rules module installed to use the Preference Service and Business Conditions within Rules.

The Rules Engine uses the information added by the Preference Service to evaluate Business Conditions, and matches only Rules whose conditions are met by the information in the request.

The Rules Engine evaluates Rules with matched Business Conditions and Rules that do not specify Business Conditions for language conditions, and eliminates Rules with Business Conditions that do not match.

See [Chapter A, Configuring Access to Business Data](#) for more information on the Preference Service and accessing session and user context information.

### EVALUATING LANGUAGE CONDITIONS

Language Conditions are conditions within a Rule that match the content of the user request.

**Note:** The Rules Engine evaluates Language Conditions only for Rules having true or no Business Conditions. See [Evaluating Business Conditions on page 32](#) for more information.

You set Language Conditions within Rules to specify actions based on the language content of the user request. The content that you can configure to match includes actual words and phrases, as well as, the more general meaning or intent. You can specify Language Conditions in the form of question patterns specified in IML, as described in Question Patterns

The Language Analyzer performs additional analysis on various semantic elements (words and phrases) as requested by the Rules Engine during this phase.

## Defining Response Criteria

The Rules Engine defines response criteria by creating a list of the actions specified by the Rules that are true for the request. Rules can specify one or more of several types of actions, as described in [Types of Actions on page 33](#).

The list of actions associated with a particular request constitutes the criteria for determining:

- The information retrieval methods and strategy
- The supplemental responses displayed in the Personalized Response User Interface answer portlets

### TYPES OF ACTIONS

Rules can specify both information retrieval actions and actions that affect how the Rules Engine processes the current Rule or subsequent Rules. Information retrieval actions include:

- Searching the indexed content in the Content Store
- Executing a structured data query
- Displaying custom content
- Displaying glossary definitions

Rule processing actions include:

- Setting an attribute for use in Business Conditions
- Restarting the request-response process with a new question

See [Types of Actions on page 33](#) for more information on the types of information retrieval actions you can specify.

## Evaluating Actions

The Evaluator analyzes the actions in the action list by ranking the responses they produce by their response scores. The high-scoring results represent the best available responses to the user request, which are displayed in the appropriate display area of the User Interface.

The Oracle Knowledge Personalized Response User Interface displays responses in the various portlets based on the answer purpose specified in the Rule that generated the response. See Answer Purposes for more information on specifying answer purposes in Rules.

## The Scoring Process

Response scoring begins when the Evaluator processes the action list associated with a request. The action list can contain a large number of actions, including Search Components that specify multiple search criteria.

During action evaluation, the Evaluator considers every word in the indexed content as a potential match for the search criteria contained in the action list. A question generally has multiple responses, and each response that the Evaluator identifies has an associated document excerpt, which is the compiled text that the User Interface displays as an answer on the results page.

**Note:** Frequently accessed index data is maintained in cache memory, as is common practices for information retrieval systems.

To determine response scores for results, the Evaluator:

- Matches the search criteria to the words within the indexed content, and generates a score for each associated excerpt, as described in [Access to Preference Data on page 182](#)
- Evaluates document metadata associated with each excerpt to determine document recency and document relevance scores, as described in [The Document Relevance and Recency Scores on page 34](#).
- Uses a weighted averaging technique to combine these elements into a final score, as described in Relative Weights.

Each excerpt that the application generates as a response receives a score. Each response's score is made up of several score values, which the application combines and weighs to calculate the total response score. The response score range is from 0 to 1.0, with 1.0 being the highest possible score.

## Search Component Scores

Each Search Component has a specified score value, which is an indicator of its value in relation to the condition with which it is associated.

Each Search Component range is associated with one or more excerpts defined by the Excerpt Service. A given excerpt can contain multiple search components of varying values.

The Evaluator calculates a score for each excerpt based on:

- The scores of the Search Components that the excerpt contains
- The proximity of the Search Components within the excerpt

Excerpts that contain enough high value search components are considered candidate responses. The Evaluator then calculates the document relevance and recency scores for the candidate excerpts.

## The Document Relevance and Recency Scores

Each excerpt is associated with a document. During content processing, the application indexes information about:

- Each document's popularity and relevance within the collection.

The relevance value for a document is calculated based on link analysis and density of the search terms. This value cannot be set manually. You can see a document's relevancy score in the Unstructured results in Test Drive when it is returned as a search result. Launch Document Search type rules use this value to set the relative weights of relevance and search components. The default weight for relevance is 0.000001% in the installed Launch Document Search Rule.

- The last update to the document.

The Evaluator assigns document relevance and recency scores based on this information. The Recency score relies on the date/time stamp of a file. The relevance and recency scores are combined and weighed with the Search Component score to determine the total response score. The relative weights for response sub-scores are determined by global settings in the application configuration, as described in the [Oracle Knowledge Intelligent Search Administration Guide](#), or by settings specified within Rules that override the global settings for this response as describe in [Add Relative Weights on page 8](#).



**Note:** This relevance weight should always have very low importance since it functions as a tie breaker in finding default search results. For the size of the content searched on a given site, factors like link analysis don't indicate relevance as reliably as they do when searching the whole web. The most important factor is the search components, which match words in the query with words in the content.

## Evaluating Anaphora

Oracle Knowledge can process requests in which an important concept that occurred in a previous request is replaced by a substitute word or phrase. The term *anaphora* refers to the relationship between the grammatical substitute (usually a pronoun such as *it*) and the previously occurring concept.

During language analysis, Oracle Knowledge resolves anaphoric expressions by matching the substitute word or phrase with the correct concept occurring in a previous request, and annotating the current request with the concept information for its predecessor.

For example, an initial request:

Where can I buy a <product>?

might be followed by:

Does it come in blue?

The Language Analyzer processes anaphoric expressions by:

- Creating a list of candidate concepts, or *salience list*, for each user request, as described in [Creating the Salience List on page 35](#)
- Determining that a word or phrase refers to a concept that occurred in a previous question, as described in [Identifying Anaphoric Expressions on page 36](#)
- Matching the referring expression to the correct concept, as described in [Resolving Anaphoric Expressions on page 37](#)
- Annotating the relevant portion of the request for use by the Rules Engine, as described in [Annotating the Anaphoric Expression on page 37](#)

## CREATING THE SALIENCE LIST

The Language Analyzer creates a salience list for each request that it processes. The salience list contains concepts that occur within the question that are potentially useful for resolving anaphoric expressions within subsequent requests.

**Important!** The Language Analyzer does not perform anaphora resolution for the first question in a session, as there is no previous question, and therefore, no salience list.

The Language Analyzer adds Concepts to the salience list if:

- The Concept is part of a noun phrase
- The Concept is not a determiner or a pronoun

**Note:** Pronoun Concepts are added to the salience list only when the Rules Engine has annotated a Concept from a previous anaphora resolution. This level of annotation enables the Language Analyzer to resolve anaphoric expressions within requests that are twice or more removed from the original Concept occurrence.

- The Concept priority level exceeds a minimum threshold. See Concept Priority Levels for more information on Concept priority levels.



The Language Analyzer partially orders the candidate concepts within the salience list using the following criteria:

- Search Component score. See [The Scoring Process on page 33](#) for more information on search component scoring.
- Concept priority level
- Sentence position (Concepts having equal Search Component scores and Concept priority levels are ordered according to their position in the request)
- Range length (concepts associated with a longer range of words within the request are ranked higher)

**Note:** The primacy of the Search Component score is based on the probability that concepts belonging to industry and application domains, which usually have higher Search Component scores, are preferred candidates for anaphora resolution.

The Language Analyzer stores the salience list with the session information so that it is available to the Rules Engine for processing the next request.

## IDENTIFYING ANAPHORIC EXPRESSIONS

The Language Analyzer identifies referring expressions using the InQuira Match Language (see [Chapter 4, IML Reference](#) for more information on IML). The anaphora-specific IML expressions match referring expressions, and eliminate non-referring pronoun expressions by testing each word in the question against the following criteria:

Expression	Definition	Examples
<b>it (pronoun)</b>	The word <i>it</i> , except when used existentially, as in the phrase <i>is it true that...</i> or <i>what does it mean when...</i>	Does <code>it</code> come in blue?
<b>deictics</b>	Words that are direct indicators. Currently defined deictic words are: <ul style="list-style-type: none"> <li>• <code>this</code></li> <li>• <code>that</code></li> <li>• <code>these</code></li> <li>• <code>those</code></li> </ul>	Where can I buy <code>this</code> ? Are there any cheaper than <code>these</code> ?
<b>deictic noun phrases</b>	Noun phrases formed from the defined deictics.	Do you have <code>this product_type</code> in blue?

When the Language Analyzer identifies a referring expression, it:

- Determines the head of the expression, as described in [Determining the Head of the Anaphoric Expression on page 36](#)
- Tests whether the referring expression can be resolved within the current request, as described in [Testing Referring Expressions Within the Current Request on page 36](#)

## DETERMINING THE HEAD OF THE ANAPHORIC EXPRESSION

The Language Analyzer determines the head of the referring expression. For single pronouns, the head of the expression is the pronoun itself. For noun phrases, for example, `the car`, the Language Analyzer determines `car` to be the head.

## TESTING REFERRING EXPRESSIONS WITHIN THE CURRENT REQUEST

The Language Analyzer checks whether the referring expression can be resolved within the current sentence by:

- Creating a salience list for the current request
- Checking that each member of the salience list occurs prior to the referring expression (with no overlap)
- Checking that all semantic constraints apply

If the Language Analyzer locates a valid concept within the current sentence's salience list, it uses that concept to resolve the referring expression, and anaphora resolution is complete. For example, in the following request:

In the <product>, does it come with air conditioning standard?

the pronoun *it* resolves to the concept associated with <product>.

## RESOLVING ANAPHORIC EXPRESSIONS

The Language Analyzer resolves anaphoric expressions by checking the previous request's salience list to locate any concepts that fit semantically with the referring expression. It checks each noun phrase candidate concept (such as *the car* or *this car*) in order, for semantic constraints. If the semantic constraints fail, it then checks the next list member.

The Language Analyzer locates concepts for anaphora resolution using concept relations defined in the Ontology. The Language Analyzer uses internally defined methods to locate the appropriate *type of* or *part of* relation for a referring expression. For example, the expression *the car* refers to a specific type of car, *model\_name*, that occurred in the previous request; and the expression *the wheel* refers to a part of the car, *model\_name*, that occurred in the previous request. See [Oracle Knowledge Language Analysis on page 30](#) for more information.

A candidate concept can resolve only one referring expression in a request. The Language Analyzer removes concepts from the list once they are used to resolve an anaphoric expression.

## ANNOTATING THE ANAPHORIC EXPRESSION

The Language Analyzer completes the anaphora resolution process by applying the #ANAPHORA label over the range of the request that contains the original referring expression. See [Language Analysis Labels on page 168](#) in the *Oracle Knowledge Language Tuning Guide* for more information on the types of labels that the Language Analyzer assigns during language analysis.

# Process Wizards

You can create custom dialogs and implement them as answers that display in response to specified questions using Oracle Knowledge Process Wizards. A Process Wizard is a decision tree that walks the user through a series of questions to either find the solution to a problem or to narrow the search for the answer to their original question. The Process Wizard feature automatically responds to specified user requests by presenting a sequence of steps that:

- Solicit more specific information from users
- Present the specific information they are seeking
- Optionally perform a search based on the information provided through the dialog

The User Interface presents Process Wizard dialogs as answers to specified questions based on custom dictionary rules that you define. When users ask questions that match Process Wizard rules, the User Interface presents a specially formatted answer on the answer page. When users select the process wizard answer, the User Interface presents the Process Wizard steps, and uses the information in the user's responses to present the desired information or to refine the initial query and perform an additional search.

For example, you could create a Process Wizard about billing problems consisting of multiple steps designed to diagnose and present solutions based on additional information provided by the user, and create a Dictionary rule to present the process as an answer in response to questions about billing problems.


When users submit a request that matches the process rule, for example, a request containing the phrase retrieve messages:

## Find Answers

The User Interface displays a special Process Wizard answer at the top of the answer page:

## Find Answers

## Answers

 **Help Retrieving Messages**  
Having problems retrieving messages? Let us guide you through it.

When a user selects the Process Wizard answer, the User Interface displays the initial step of the process that you have created to diagnose and resolve the user's specific problem related to retrieving messages:

<a href="#">» Start over</a>	<b>Help Retrieving Messages</b> Having problems retrieving messages? Let us guide you through it.
<b>Your Responses</b> What kind of message are you trying to retrieve? <b>Email Message</b> From where are you trying to retrieve your messages? <b>From your home computer</b>	To read e-mail in your online mailbox  1. Click the Read icon on the toolbar. 2. On the New tab, double-click the first e-mail item to read it.  <div> <input type="button" value=" &lt; Back"/> <input type="button" value=" Finish"/> </div>

## Implementing the Process Wizard Feature

You implement Process Wizards within your application by:

- Creating one or more process wizards using the Process Wizard Editor application of the Oracle Knowledge Language Workbench, as described in [Creating Process Wizards on page 77](#).
- Creating or modifying Dictionary rules to invoke the process wizard in response to user requests, as described in [Creating Dictionary Rules to Invoke Process Wizards on page 95](#).

The default User Interface contains page templates to display process wizard answers, as described in [The Process Wizard User Interface on page 96](#); no User Interface customization is required to implement Process Wizards.

**Important!** For quick assistance in creating and implementing a basic process wizard, see the tutorial, [Tutorial: Creating a Process Wizard on page 57](#), which describes in detail how to create a process wizard to assist users in retrieving messages from a hypothetical email, voice, and instant message system.

## Process Wizard Elements

A Process Wizard contains various elements that you create and define using the Process Wizard Editor. Elements within a process include:

- Process steps, which are the individual modules that are presented to the user in the Process Wizard User Interface, as described in [Process Wizard Steps on page 43](#).
- Fields within the process steps, which contain the text and user option elements that provide information and solicit user responses, as described in [Process Wizard Fields on page 43](#).
- Conditions and transitions, which are the logical framework for determining the navigation through the process based on the options selected by the user, as described in [Process Wizard Conditions and Transitions on page 43](#).
- The process wizard User Interface components, which include a special answer format, and special Process Wizard presentation pages, as described in [Process Wizard User Interface Elements on page 44](#).

## Process Wizard Steps

A Process Wizard consists of any number of process steps that you design and create to:

- Solicit information from end users
- Guide end users to specific information

Process steps are individual components that consist of any number of fields that you can define. Fields within process steps include text fields to provide information and option fields that display options or accept input from users as they progress through a process.

See [Creating Process Steps on page 79](#) for more information.

## Process Wizard Fields

Fields within process steps include text fields to provide information and option fields that display options or accept input from users as they progress through a process.

You can define individual fields that are specific to a process as well as shared fields that you can re-use in multiple processes or use as templates to easily create variants of common types of fields.

See [Defining Fields on page 84](#) and [Creating Shared Fields on page 89](#) for more information.

## Process Wizard Conditions and Transitions

You connect process steps to one another by specifying conditions and transitions. Conditions correspond to the various user options presented within a step, and transitions are the method of connecting to the next logical step in the process based on a selected condition.

You define conditions and transitions by:

- Creating a condition for each user option in a step
- Associating a transition with each condition

See [Defining Conditions on page 93](#) and [Defining Transitions on page 94](#) for more information.

## Process Wizard User Interface Elements

The process wizard User Interface components include the special answer format that displays in the standard answer area when a user request matches the process wizard rule, the pages that correspond to the steps within the process, and a side-bar that summarizes users' responses as they progress through the process wizard.

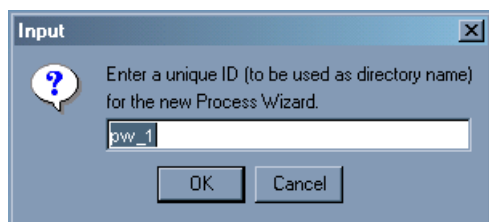
See [The Process Wizard User Interface on page 96](#) for more information.

## Accessing the Process Wizard Editor

You access the Process Wizard Editor from the Dictionary Manager application of the Oracle Knowledge Language Workbench, using any of these methods:

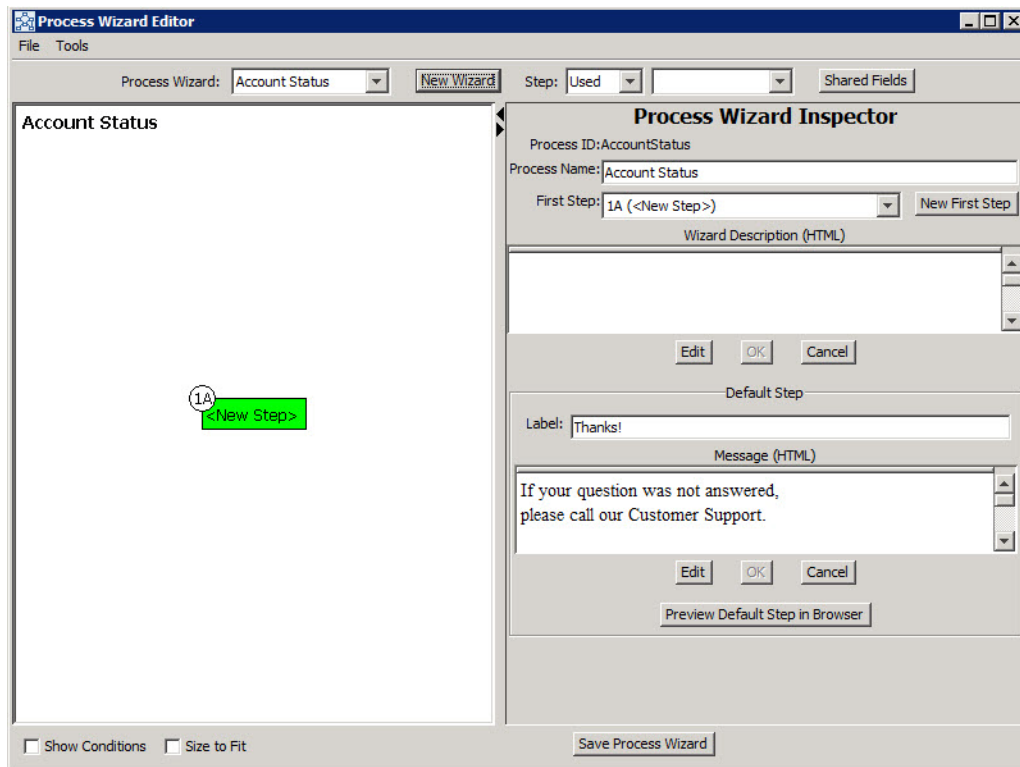
- 1 Select the Process Wizard icon , on the Dictionary Manager toolbar
- 2 Select **Process Wizard Editor** from the Dictionary Manager Tools menu
- 3 Select **Display Process Wizard Editor** from the Tools menu of a Dictionary rule

If there are no process wizards currently defined, the Process Wizard Editor displays a dialog that prompts you to create a directory to contain the initial process wizard information:



- 4 Enter a directory name, or accept the default value (pw\_1), and select **OK**

The Process Wizard Editor displays a new process wizard, or the most recently accessed process wizard if any exist.



See [Using the Process Wizard Editor on page 46](#) for more information.

## Using the Process Wizard Editor

The Process Wizard Editor displays in a separate window from the Dictionary Manager.

**Note:** You can close the Process Wizard Editor without affecting the Dictionary Manager; however, closing the Dictionary Manager also closes the Process Wizard Editor.

The Process Wizard Editor display is divided into two sections:

- The Graphic View, as described in [The Graphic View on page 47](#)
- The Inspector View, which initially displays the Process Wizard Inspector, and which you can toggle to display the Process Step Inspector and the Shared Fields Inspector. (See [The Process Wizard Inspector on page 50](#), [The Process Step Inspector on page 51](#), and [The Shared Fields Inspector on page 56](#) for more information.)

You can change the relative sizes of the graphic view and wizard inspector by selecting and dragging the dividing bar to the left or right as needed. You can expand either of the areas completely by selecting the appropriate arrow icon located on the divider.

The Process Wizard Editor displays the following functions above the graphic view and wizard inspector:



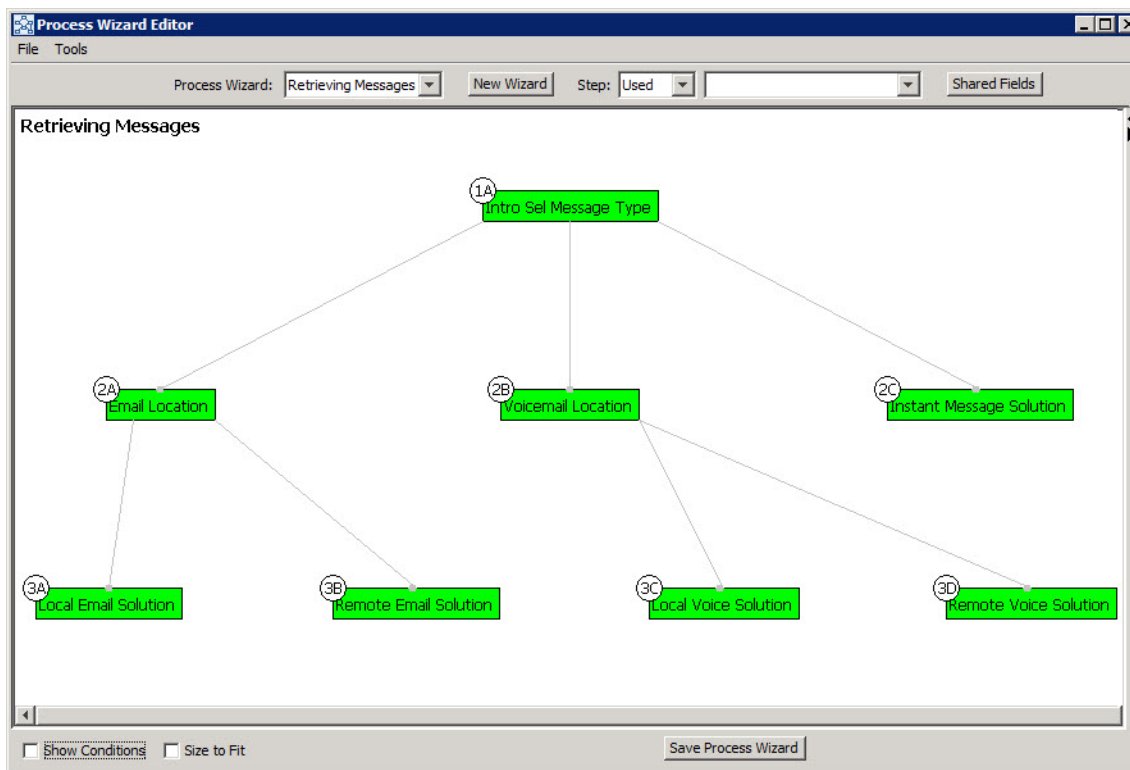
Function	Description
<b>Process Wizard</b>	Use this to select an existing process wizard. The Process Wizard Editor selects the most recently accessed wizard by default.
<b>New Wizard</b>	Use this to create a new Process Wizard.
<b>Step</b>	Use this field to display any of the defined process steps from the used and unused steps lists. Used steps are those that are included in the current process. Unused steps are defined steps that are not currently included in the process.
<b>Shared Fields</b>	Use this field to view, create, and modify fields that are intended for use in multiple processes, as described in <a href="#">The Shared Fields Inspector on page 56</a> and <a href="#">Creating Shared Fields on page 89</a>



## The Graphic View

The Process Wizard Editor graphic view displays the selected process wizard as an arrangement of:

- Labeled boxes representing the process steps
- Lines connecting the steps representing the transitions, for which you can also display labels indicating the condition associated with each transition.



The steps are arranged within the graphic view as a tree, starting with the initial step. The graphic view arranges the subsequent steps into rows; the row in which a step is displayed is determined by its transition relationships; each step is displayed below all steps that transition to it.

**Note:** The layout of the steps and connections within the graphic view is automatically determined by the data in the process inspector; you cannot directly edit the graphic view.

The Process Wizard does not support loops within solution paths.

The graphic view displays each process step as a rectangle, labeled with:

- The name of the step
- The step ID



You can adjust the display size of the graphic view to fit more process steps into the screen using the Size to Fit checkbox located near the bottom of the Process Wizard Editor display.

## Step Displays in the Graphic View

When the Process Wizard Inspector is active in the inspector view (see [The Process Wizard Inspector on page 50](#)), the graphic view displays the process steps in cyan (blue-green). If you select a process step, the inspector view displays the Process Step Inspector (see [The Process Step Inspector on page 51](#)) and the graphic view displays the selected step in yellow.

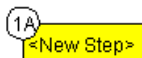
You can select a step within the graphic view to display its data in the Process Step Inspector:

- Select (click on) a step in the graphic view



The Process Wizard Editor:

- Displays the step in yellow within the graphic view



and

- Displays the step details in the step inspector

Step ID: 1A

Step Name: <New Step>

Step Fields

Step Transitions

Step Preview

Select	Type	Field Name (for Conditions)	Actions
<input type="checkbox"/>	html		Delete

☐ Shared Field

Description for Selected Field

Description (HTML)

\*\*\* Replace with step description \*\*\*

Edit

OK

Cancel

Preview Step In Browser

Delete Step

**Note:** You can toggle to the Process Wizard Inspector by selecting (clicking) anywhere in the background of the graphic view when the Process Step Inspector is active.

## PROCESS WIZARD EDITOR STEP IDENTIFICATION

The Process Wizard Editor maintains unique IDs for each process step. The step ID is composed of:

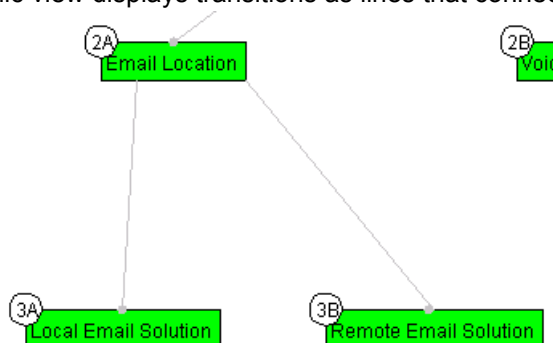
- The row number, which can be in the range of 1-99
- A letter that corresponds to the step's column position within the row (A-Z)

For example, the third step in the second row has the ID: 2C.

**Note:** The step ID is based only on the step's position within the process, not its contents; moving a step from one position in the process to another changes the step ID. The Process Wizard Editor automatically arranges the steps within the process as you create or delete steps, conditions, and transitions.

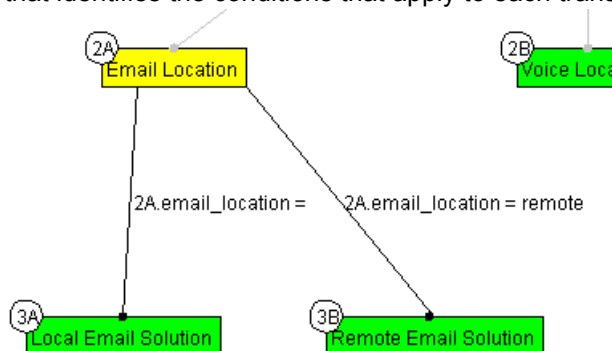
## Conditions and Transitions in the Graphic View

The graphic view displays transitions as lines that connect the process steps.



When you select a step, the Process Wizard Editor displays:

- Bold transition lines from the selected step
- Text that identifies the conditions that apply to each transition



You can display all of the conditions for the process using the Show Conditions field located near the bottom of the Process Wizard Editor display.

## The Process Wizard Inspector

The Process Wizard Inspector displays detailed information about the current process wizard.

The Process Wizard Inspector displays the following fields and functions:

Field	Description
<b>Process ID</b>	Indicates the internal name of the process wizard.
<b>Process Name</b>	Specify the name of the process wizard. The User Interface displays the contents of this field as the title of the link to the process wizard.
<b>First Step</b>	Indicates the current first step in the process, and lists the additional steps. You can change the first step using the First Step drop-down menu or the New First Step button, as described in <a href="#">Creating a New First Step on page 80</a> and <a href="#">Designating a New First Step on page 81</a> .
<b>Wizard Description</b>	Specify an optional description of the process wizard. The application displays the contents of this field as additional text in the link to the process wizard.

The Process Wizard Inspector also displays information about the Default Step, as described in [The Default Step on page 51](#).

## The Default Step

Each process wizard includes a Default Step, which is a generic step that the Process Wizard displays when:

- There is a gap in the process created by a condition for which no transition is defined
- The user selects Next, without responding in a way that satisfies any of the step's conditions

You can customize the default step by editing the following fields:

Field	Description
<b>Default Step Label</b>	Use this field to specify the name to display in the graphic view if this step is explicitly specified in a transition. The default value is Thanks !
<b>Default Step Message</b>	Use this field to specify plain text or HTML that the User Interface displays to the end user. The default value is:  If your question was not answered, please call our Customer Support

**Note:** The Default Step does not support user options, conditions, transitions, or the Next button.

You can display a preview of the default step using the Preview Default Step in Browser button. The Default Step displays in a separate browser window.

## The Process Step Inspector

You can view and modify the details of a process step using the Process Step Inspector.

You access the Process Step Inspector by selecting a step:

- From the Graphic View
- From either of the drop-down Steps menus at the top of the Process Wizard Editor

The Process Step Inspector displays the following functions and fields:

Function	Description
<b>Step ID</b>	Displays the step ID, as described in <a href="#">Process Wizard Editor Step Identification on page 49</a> .
<b>Step Name</b>	Displays the step name, which is used only within the Process Wizard Editor.
<b>Step Fields tab</b>	Displays details for the fields defined for the selected process step, as described in <a href="#">The Step Fields Tab on page 53</a> .
<b>Step Transitions tab</b>	Displays details for the transitions defined for the selected process step, as described in <a href="#">The Step Transitions Tab on page 54</a> .
<b>Step Preview tab</b>	Displays a preview of the selected process step, as described in <a href="#">The Step Preview Tab on page 55</a> .
<b>Preview Step in Browser</b>	Click to display the current step as it is to be displayed in the default User Interface. The Process Wizard Editor displays the step in a separate browser window, as described in <a href="#">Previewing Steps within a Browser on page 83</a>

## The Step Fields Tab

You can view and modify the fields within a process step using the Step Fields tab of the Process Step Inspector. You can define various types of fields to provide information, present options, and accept input from users as they progress through the steps in a process.

The Step Fields table displays one field on each line, and the last line of the table is always blank to accept input to define a new field.

You define a field by specifying the following parameters, as described in [Defining Fields on page 84](#):

Field	Description
Type	Use this field to specify the field type, which can be a unique instance of a field, or an instance of a shared field. The field type determines the functionality of the field, as described in <a href="#">Specifying Field Types on page 86</a> .

<b>Field Name (For Conditions)</b>	Use this parameter to specify a unique ID to use to refer to the field when specifying Conditions and Transitions, as described in <a href="#">Defining Conditions on page 93</a> .
<b>Description</b>	Use this field to specify: <ul style="list-style-type: none"> <li>• markup and text for html fields</li> <li>• markup and text for user option fields (checkbox, radio, select, text)</li> </ul>

## The Step Transitions Tab

You can view, create, and edit the transitions defined for a step using the Step Transitions tab. Transitions determine the steps that are presented to users based on their response to one or more user options.

The Step Transitions table displays one transition on each line, and the last line of the table is always blank to accept input to define a new transition.

Each transition corresponds to a condition; if you specify a transition that refers to multiple conditions, all of the conditions must be true for the transition to occur (an AND relationship). You can specify that a transition occurs based on any of multiple conditions being true (an OR relationship) by adding multiple transitions to the same step.

You can add transitions to existing steps, or create a new step for a transition.

See [Defining Transitions on page 94](#) for more information on using the Step Transitions fields and functions.

The Step Transitions table displays the following fields for defining transitions and conditions:

Field or Function	Description
<b>Go To Step</b>	Use this field to select an existing step as the destination step for this transition.
<b>Actions</b>	Use the action <b>Delete</b> to delete the transition to its left. Use the action <b>New Step</b> to create a new step as the destination step for this transition.

You can drag and drop the transitions using the mouse on the Select column, to rearrange the transitions in the Graphic View.

### Conditions for Selected Transition

<b>Field Name</b>	Use this field to specify the condition for the transition.
<b>Op (Operator)</b>	Use this field to specify the operator to use as the basis of comparison for the user response and the specified value.
<b>Value</b>	Use this field to specify the value to compare the user response to evaluate the condition

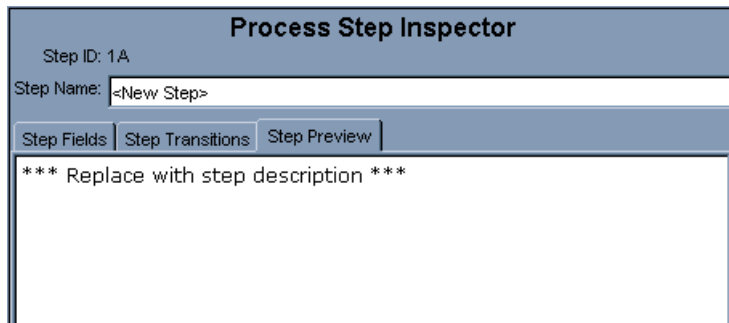


## The Step Preview Tab

You can display a draft view of the current step using the Step Preview tab.

The Step Preview tab displays a general HTML rendering of the step fields; however, it is not as complete as the preview displayed in a separate browser window, as described in [Previewing Steps within a Browser on page 83](#).

The Step Preview tab provides an efficient way to verify the text and options of all the steps in a process. For example, you can select the Step Preview tab, then select each step in the graphic view.



The screenshot shows the 'Process Step Inspector' dialog box with the 'Step Preview' tab selected. The 'Step ID' is '1.A' and the 'Step Name' is '<New Step>'. The main area displays '\*\*\* Replace with step description \*\*\*'.

Process Step Inspector		
Step ID: 1.A		
Step Name: <New Step>		
Step Fields	Step Transitions	Step Preview
*** Replace with step description ***		

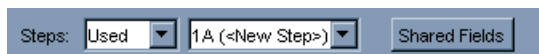
## The Shared Fields Inspector

You can view, create, and manage step fields that can be shared among multiple steps and processes using the Shared Fields Inspector. Shared fields enable you to create and manage a central library of commonly used fields and templates for use within process steps.

You can use shared fields:

- As standard fields shared among multiple process wizards
- As templates for creating similar fields

You access the Shared Fields Inspector from the Shared Fields button at the top of the Process Wizard Editor window:



The Shared Fields Inspector lists all of the shared fields defined within your application. The last row of the table is always blank to accept new shared field definitions.

Shared Fields Inspector			
Select	Type	Shared Field Type (for Step Inspector)	Actions
	html	about_help_wizards	Delete
	radio	sel_wizard_action	Delete
	checkbox	checkbox_2	Delete
Description for Selected Field			
Description (HTML)			
<div> <input type="button" value="Edit"/> <input type="button" value="OK"/> <input type="button" value="Cancel"/> </div>			
Options for Selected Field			
Value Name (for Conditions)	Value	Default	Actions Up Dn
		<input type="checkbox"/>	

The Shared Fields table contains the following fields:

Field	Description
Type	Use this field to specifying the generic type of the shared Field.
Shared Field Type (for Step Inspector)	Use this field to specify a unique ID to use to refer to the field when adding the field to a process step.
Description for Selected Field	Use this field to specify: <ul style="list-style-type: none"><li>• markup and text for html fields</li><li>• markup and text for user option fields (checkbox, radio, select, text)</li></ul>
Value Name (for Conditions)	Use this field to specify the value name for the option for use in specifying conditions and transitions.
Display Value	Use this field to specify the value that displays for the option.
Default	Use this checkbox to specify that this option is pre-selected when this field is displayed to the user.

See [Creating Shared Fields on page 89](#) for more information.

## Tutorial: Creating a Process Wizard

This section guides you through the basic steps of creating a process wizard. This tutorial provides:

- Step-by-step instructions to design and implement a process for a hypothetical scenario
- Explanations of the selected options
- References to more detailed information about additional options and features

## Scenario: Creating a Process to Assist Message Retrieval

In this scenario, we create a simple process to help end-users retrieve messages from a hypothetical service that provides:

- Email
- Voice mail
- Instant messaging

In addition, the process also needs to account for the fact that users have the capability to retrieve voice and email messages from:

- Their home location (computer or telephone)
- A remote location

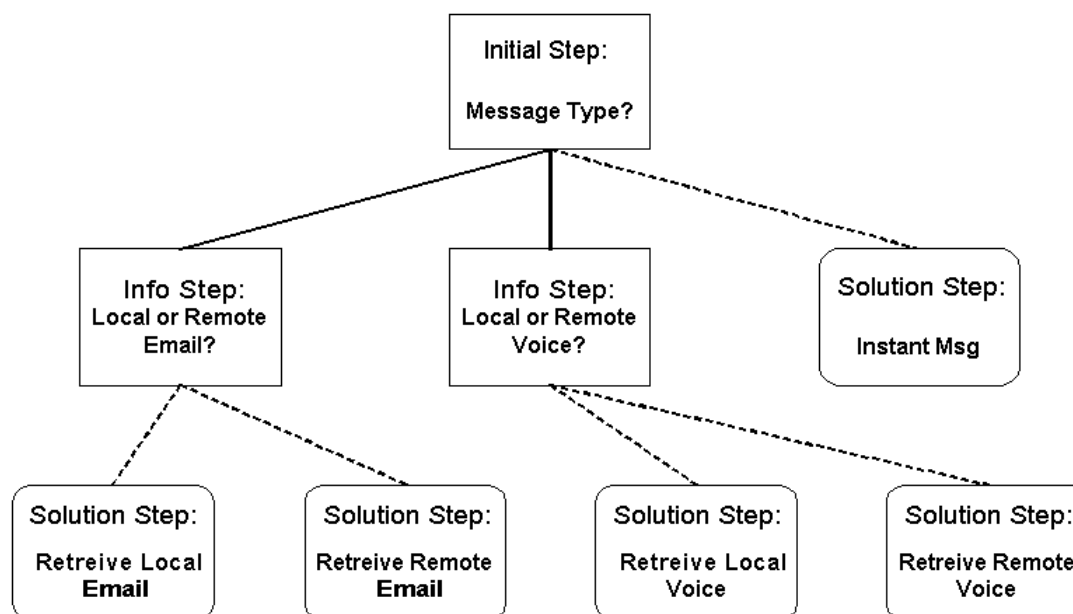
To assist users in retrieving any type of message, from either location, we create a process consisting of steps to:

- Determine the type of message the user is trying to retrieve
- Determine the location from which the user is trying to retrieve email or voice mail
- Present the appropriate solution based on the responses to the user options

**Note:** [Tutorial Review on page 76](#) contains an image of the completed process wizard as displayed in the graphic view for your reference.

## Designing the Wizard

To begin the process wizard we can design the process by creating a diagram showing the steps and transitions that we want to include in the process:



**Note:** The process diagrams in this section are similar to, but not intended to represent exactly, the graphic view displayed within the Process Wizard Editor.

[Designing the Process Steps on page 59](#) describes the types of steps that are included in the process and their arrangement within the process diagram.

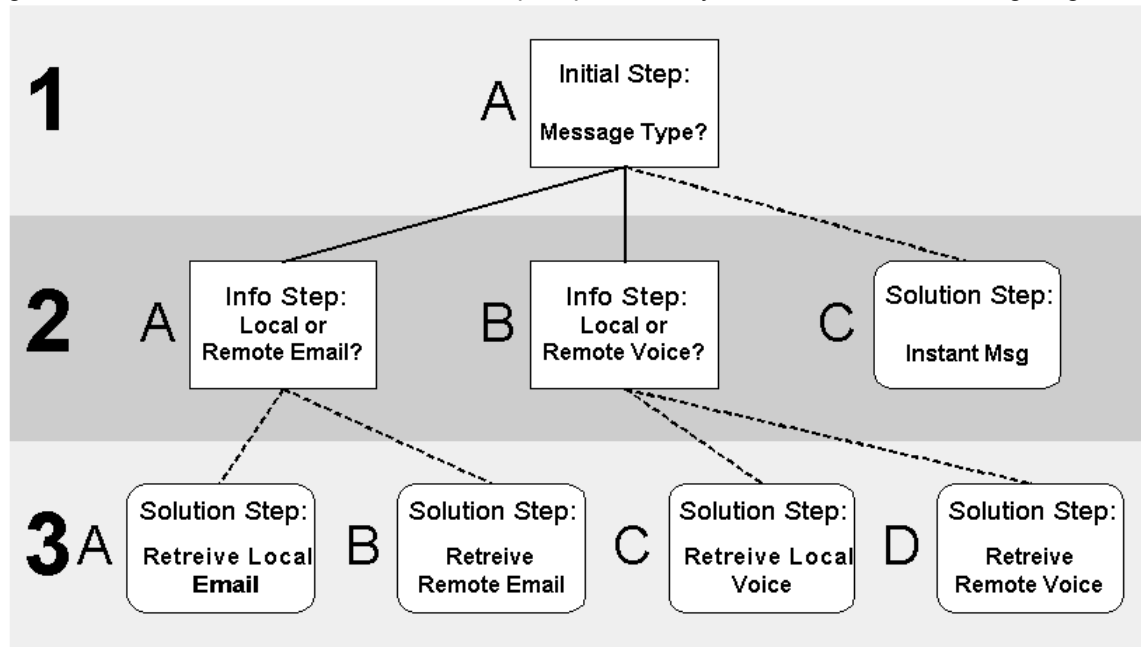
## Designing the Process Steps

The process design diagram shows three types of process steps:

- The initial step, which is the entry point into the process wizard. The initial step is displayed when a user selects a process wizard answer from the User Interface answer page. See [Defining the Initial Step on page 64](#) for more information.
- Information steps, which are intermediate steps that prompt the user for required information
- Solution steps, which provide the appropriate information or action in response to the information entered by the user in previous steps

The process steps are arranged as a set of branching steps, connected to one another by lines that represent the transitions based on the conditions defined within each step. Each set of connected steps represents a path from the entry point to a solution.

Since there is a maximum of three steps from the entry point to a solution, we can label the rows of the diagram 1-3. Within each row, we label the steps alphabetically, as shown in the following diagram:



This labeling scheme is used by the Process Wizard Editor when arranging the steps defined within a process.

[Designing Step Fields on page 59](#) describes the basic process of designing the text and user options that you define as fields within the various process steps.

## Designing Step Fields

Each process step can contain multiple elements, such as introductory text, lists or menus of available choices, and prompts for values. We add these elements as individual fields within the process steps that we create.

For example, we might want our initial step to contain:

- Introductory text about the process wizard and the nature of the problem it is designed to solve

- Text that prompts the user to specify the type of messages that they need assistance in retrieving
- A set of radio buttons from which the user can select the appropriate type of message

The following table shows examples of these elements:

Field Type	Example
HTML Text	If you are having trouble retrieving your messages, this wizard guides you to the right solution. Please take a moment to answer a few questions, so that we can provide the right information to help you retrieve your messages.
Radio Button Intro Text	What type of message are you trying to retrieve?
Radio Buttons	<ul style="list-style-type: none"> <li>• Email Message</li> <li>• Voice Message</li> <li>• Instant Message</li> </ul>

The following graphic is an example of the step as we display to the end user:

If you are having trouble retrieving your messages, this wizard will guide you to the right solution. Please take a moment to answer a few questions, so that we can provide the right information to help you retrieve your messages.

**What kind of message are you trying to retrieve?**

☒ Email Message  
☐ Voice Message  
☐ Instant Message

**Note:** The Back and Finish buttons are automatically included within each step. When we add transition steps from the initial step, the Next button automatically displays in place of the Finish button.

[Designing Step Conditions and Transitions on page 60](#) describes how to design the conditions, which correspond to the user options, and transitions, are the destination steps that correspond to each condition.

## Designing Step Conditions and Transitions

The steps in our process are connected to one another by conditions, which correspond to the possible user responses to the prompts within the process steps. Each condition is associated with a transition, which determines the next step that displays in the process.

The following tables show the conditions and transitions that we create for our process. Each table represents a row in the process diagram displayed in [Designing the Process Steps on page 59](#). For each step within the row, the Condition column lists the available user options and their transition steps.

**Note:** The steps labeled Solution Step: (2C, 3A, 3B, and 3C) have no conditions and transitions, since they are the final steps in the process.

Step	Description	Condition	Transition
<b>1A</b>	Initial Step: Message Type	Email	2A - Local or Remote Email
		Voice Mail	2B - Local or Remote Voice Message
		Instant Message	2C - Instant Message Solution
<b>2A</b>	Information Step: Local or Remote Email	Local	3A - Local Email Solution
		Remote	3B - Remote Email Solution
<b>2B</b>	Information Step: Local or Remote Voice Message	Local	3C - Local Voice Message Solution
		Remote	3D - Remote Voice Message Solution
<b>2C</b>	Solution Step: Instant Message	N/A	N/A
<b>3A</b>	Solution Step: Local Email	N/A	N/A
<b>3B</b>	Solution Step: Remote Email	N/A	N/A
<b>3C</b>	Solution Step: Local Voice Msg	N/A	N/A
<b>3D</b>	Solution Step: Remote Voice Msg	N/A	N/A

[Defining the Process Using the Process Wizard Editor on page 62](#) describes how to add the process to our application.

## Defining the Process Using the Process Wizard Editor

To create the process we perform the following steps using the Process Wizard Editor:

- Define a new process, as described in [Defining a New Process on page 62](#)
- Define the initial step, conditions, and transitions, as described in [Defining the Initial Step on page 64](#)
- Define the intermediate information steps, conditions, and transitions, as described in [Defining Information Steps on page 68](#)
- Define the solution steps, as described in [Defining Information Steps on page 68](#)

See [Accessing the Process Wizard Editor on page 44](#) and [Using the Process Wizard Editor on page 46](#) for more information on accessing and using the features of the Process Wizard Editor.

## Defining a New Process

To define a new process:

- 1 Start the Process Wizard Editor, as described in [Accessing the Process Wizard Editor on page 44](#)

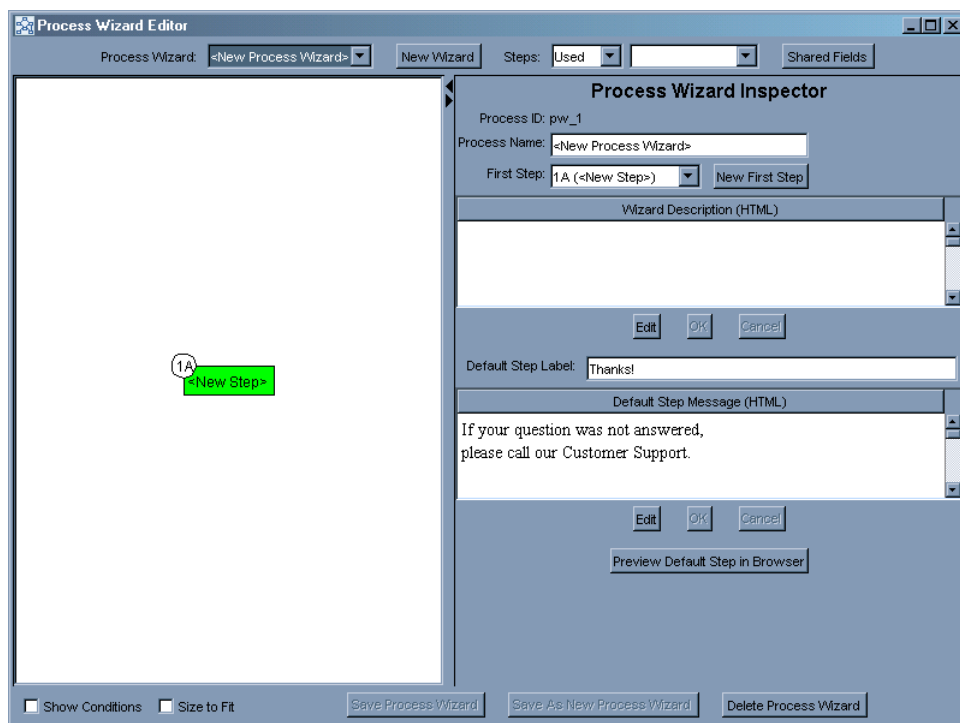
The Process Wizard Editor selects the most recently accessed process wizard. If no process wizards are currently defined, the Process Wizard Editor prompts you to name the directory for the initial process wizard data.

- 2 Enter the directory name and select **OK**



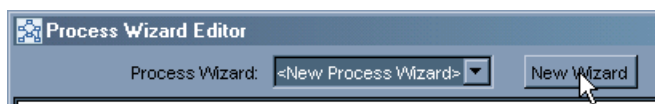


The Process Wizard Editor displays a new process wizard:



If there are existing process wizards, the Process Wizard Editor displays the most recently accessed wizard. If so, create a new process wizard:

- 3 Select the **New Wizard** button



- 4 Enter a unique ID and select **OK**



- 5 Enter the name of the process wizard in the **Process Name** field:



**Note:** You can save the new process at any time using the Save Process Wizard button located below the process diagram window.

Next, we define the initial step in the process definition area, as described in [Defining the Initial Step on page 64](#).

## Defining the Initial Step

In this section, we create the initial step, and define within it the following fields:

- Introductory text about the process wizard and the nature of the problem it is designed to solve
- A set of radio buttons from which the user can select the appropriate message type

To define the initial process step:

- 1 Select **1A(<New Step>)** the graphic view



The Process Wizard Editor displays the Process Step Inspector.

- 2 Enter a new name in the Step Label field:



**Process Step Inspector**

Step ID: 1A

Step Name: Intro Sel Message Type

We can now define the text field for our introductory text, as described in [Defining the Introductory Text Field on page 65](#).

## DEFINING THE INTRODUCTORY TEXT FIELD

We define the introductory text field using the Step Fields tab of the Process Step Inspector. The introductory text introduces the wizard to the user.

The Step Fields table contains a default entry for an html field, which we edit for our process. To edit the default entry:

- 1 Since this field is informational, there are no conditions to specify. Leave the **Name (For Conditions)** column of the default html entry blank:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete

- 2 Select the **Edit** button below the Description field, and enter the introductory text:

Description (HTML)
If you are having trouble retrieving your messages, this wizard will guide you to the right solution. Please take a moment to answer a few questions, so that we can provide the right information to help you retrieve your messages.

**Note:** You can enter any valid text formatting html in the Description field.

We can now define the user options field for the initial step, as described in [Defining the User Options for the Initial Step on page 65](#).

## DEFINING THE USER OPTIONS FOR THE INITIAL STEP

We define the user options field by creating a new field in the Step Fields table. The user options field prompts users to select the type of message for which they need assistance from a set of radio buttons.

To create a new field for the user options:

- 1 Select the empty field below the introductory text field by clicking in the **Type** column, and select **radio** from the drop-down list:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete
	html		
<input type="checkbox"/> Share	checkbox		
	html		
	radio	Description for Selected Field	
	search	Description (HTML)	
	select		
	text		

- 2 Select the **Field Name (for Conditions)** column and override the default field name by entering a meaningful field name for the condition, which prompts users to select a message type:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete
	radio	message_type	Delete

See [Defining Conditions on page 93](#) for more information.

- 3 Enter the Description for the user options field:

Description (HTML)
What type of message are you trying to retrieve?

- 4 Enter condition value names and values for the email, voice message, and instant message options

**Note:** You can use the checkbox in the Default column to specify that an option is selected by default when the step displays. For this process, we assume that the email option is the most likely user response and should therefore be selected by default.

Value Name (For Condition)	Value	Default
email	Email	Yes
voice	Voice	No
instant	Instant Message	No

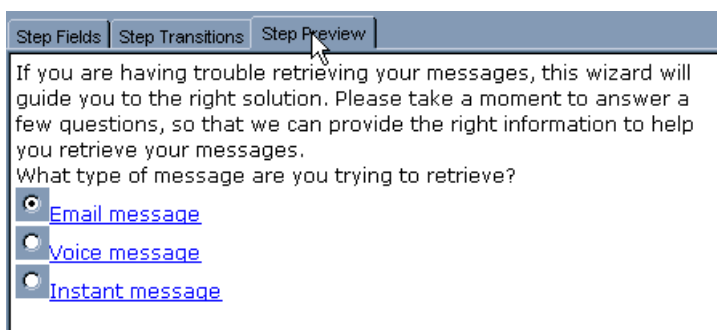
Options for Selected Field			
Value Name (for Conditions)	Value	Default	Actions
email	Email message	<input checked="" type="checkbox"/>	Delete
voice	Voice message	<input type="checkbox"/>	Delete
instant	Instant message	<input type="checkbox"/>	Delete

We can now display a preview of our initial step, as described in [Previewing the Initial Step on page 67](#), then continue defining our process by specifying transitions to the various information steps, as described in [Defining Information Steps on page 68](#).

## PREVIEWING THE INITIAL STEP

We can display a preview of the initial step:

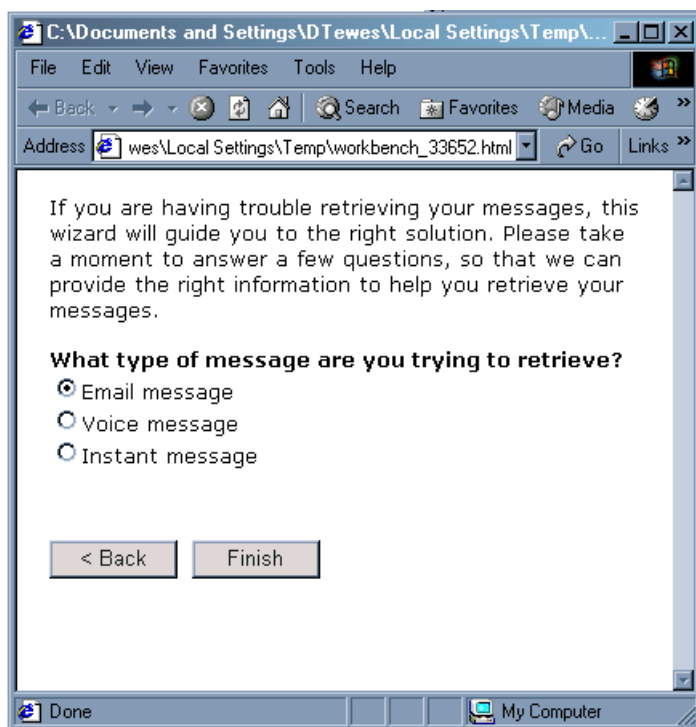
- By selecting the **Step Preview** tab of the Process Step Inspector:



- By selecting the **Preview Step in Browser** button near the bottom of the Process Step Inspector screen:



The Process Wizard Editor invokes a browser window to display the current process step:



**Note:** See [Previewing Steps on page 82](#) for more information on displaying previews of process steps.

We can now continue creating the process by defining information steps, as described in [Defining Information Steps on page 68](#).

## Defining Information Steps

In this section, we define a step that presents user options to solicit the required information from end-users. In [Defining the Initial Step on page 64](#), we created our initial step, which asks users to specify the type of message that they need help retrieving. In order to direct users to the appropriate solution for the type of message they are interested in, we need to define steps to determine whether users need assistance:

- Retrieving email messages from their home location or from a remote location
- Retrieving voice messages from their home location or from a remote location

**Note:** Users who select the instant message option transition directly to the instant message solution step, as described in Defining the Instant Message Solution Step, since there is no distinction between home and remote locations when receiving instant messages.

For this tutorial we create the email location step, which prompts users to specify whether they are trying to retrieve messages from their home location or from a remote location.

### DEFINING THE EMAIL MESSAGE LOCATION STEP (STEP 2A)

To define a step that prompts users to specify whether they need assistance retrieving email messages from their home computer (local), or from another computer (remote):

- 1 Create a new step, as described in [Creating the Email Location Step on page 69](#).
- 2 Specify the condition that we want to associate with the new step, as described in [Specifying the Condition for the Email Location Step on page 69](#).
- 3 Define the fields within the new step, as described in [Specifying the Email Location Step Name on page 70](#).

## CREATING THE EMAIL LOCATION STEP

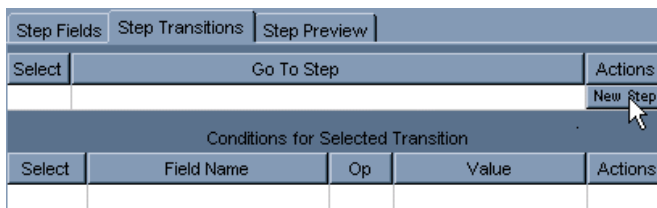
We create the email location step using the New Step function of the Process Step Inspector's Step Transitions tab:

- 1 Select the initial step (1A)



The Process Wizard Editor displays the Process Step Inspector

- 2 Select the **Step Transitions** tab and select the **New Step** button



The Process Wizard Editor creates a new step with the default name 2A (<New Step>).

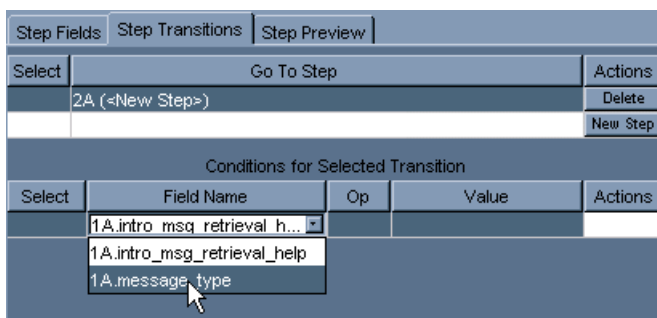
We specify the conditions for the email location step, as described in [Specifying the Condition for the Email Location Step on page 69](#).

## SPECIFYING THE CONDITION FOR THE EMAIL LOCATION STEP

To define the transition from the initial step to the new email location step, we specify the condition using the Conditions for Selected Transition table of the Step Transitions tab.

We use the Field Name and Value Names that we defined in [Defining the User Options for the Initial Step on page 65](#) to specify that the transition to the email location step occurs when users select the email option on the initial screen.

- 1 Select the **Field Name** column of the Conditions for Selected Transition table, then select the field **1A.message\_type** from the drop-down list



- 2 Select the **Value** column, then select the value **email** from the drop-down list

Conditions for Selected Transition				
Select	Field Name	Op	Value	Actions
	1A.message_type	=	email	Delete
			email	
			voice	

**Note:** The Process Wizard Editor selects the Operator (Op) = by default.

We can now specify the step name, as described in [Specifying the Email Location Step Name on page 70](#), and define the fields for the email location step.

## SPECIFYING THE EMAIL LOCATION STEP NAME

To specify the name of the email location step, we use the Step Name field of the Process Step Wizard. To define the step name:

- 1 Select the step **2A(<New Step>)** the graphic view



The Process Wizard Editor displays the Process Step Inspector.

- 2 Enter the new name in the **Step Label** field:

Process Step Inspector			
Step ID: 2A			
Step Name: Email Location			
<div>Step Fields</div> <div>Step Transitions</div> <div>Step Preview</div>			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete

We can now define the text field for our introductory text, as described in [Defining the Introductory Text Field for the Email Location Step on page 71](#).



## DEFINING THE INTRODUCTORY TEXT FIELD FOR THE EMAIL LOCATION STEP

We define the introductory text field for the email location step using the Step Fields tab of the Process Step Inspector. The introductory text asks the user to specify whether they are trying to retrieve local or remote messages.

The Step Fields table contains a default entry for an html field, which we edit for our process. To edit the default entry:

- 1 Select the default html entry

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete

- 2 Select the Edit button below the Description field, and enter the introductory text:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete
<input type="checkbox"/> Shared Field Description for Selected Field Description (HTML) Please tell us where you trying to retrieve your email.			

**Note:** You can enter any valid text formatting html in the Description field.

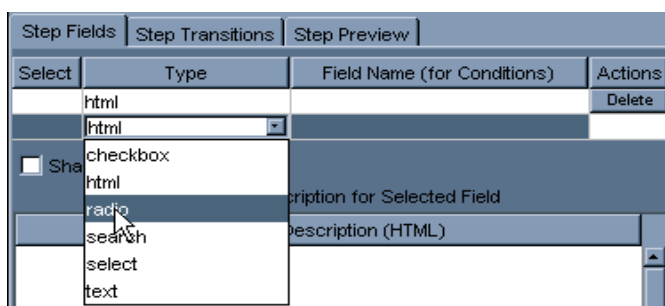
We can now define the user options field for the email location step, as described in [Defining User Options for the Email Location Step on page 72](#).

## DEFINING USER OPTIONS FOR THE EMAIL LOCATION STEP

We define the user options using the same processes described in [Defining the User Options for the Initial Step on page 65](#). The user options field for this step prompts users to select a radio button that corresponds to the location from which they are trying to retrieve messages.

To create the user options field:

- 1 Select the empty field below the introductory text field by clicking in the **Type** column, and select **radio** from the drop-down list:



- 2 Select the **Field Name** column and override the default field name by entering an appropriate field name for the condition:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete
	radio	email_location	Delete

See [Defining Conditions on page 93](#) for more information.

- 3 Enter the Description for the user options field:

Description (HTML)
What type of message are you trying to retrieve?

- 4 Enter condition value names and values for the local and remote options.

**Note:** You can specify that an option be selected by default; for this process, we assume that the local option is the most likely user response and should therefore be selected by default.

Value Name (For Condition)	Value	Default
local	My regular computer	Yes
remote	A different computer	No

Options for Selected Field			
Value Name (for Condition)	Value	Default	Actions
local	My regular computer	<input checked="" type="checkbox"/>	Delete
remote	A different computer	<input type="checkbox"/>	Delete
		<input type="checkbox"/>	

You can use the same processes to define the corresponding voice mail location step (2B) and its fields.

This tutorial continues with an example of a solution step, as described in [Defining the Local Email Solution Step on page 73](#).

## Defining Solution Steps

In this section, we define a step that presents a solution based on selected user options. In [Defining Information Steps on page 68](#), we created the email location step, which prompts users to specify whether they are trying to retrieve messages from their home location or from a remote location.

For this tutorial we create the local email solution step, as described in [Defining the Local Email Solution Step on page 73](#), which provides information about how to retrieve email from the user's regular home or office computer.

### DEFINING THE LOCAL EMAIL SOLUTION STEP

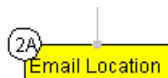
To define a solution step that provides information based on the specified user options in our process, we:

- Create a new step, as described in [Creating the Email Location Step on page 69](#)
- Specify the condition that we want to associate with the new step, as described in [Specifying the Condition for the Email Location Step on page 69](#)
- Define the fields within the new step

### CREATING THE LOCAL EMAIL SOLUTION STEP

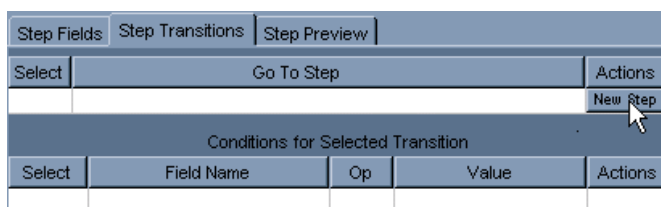
We create the email location step using the same process as used in [Creating the Email Location Step on page 69](#), except that we create a step in the third row of our process by selecting the New Step function from the email location step (2A):

- 1 Select the email location step (2A)



The Process Wizard Editor displays the Process Step Inspector.

- 2 Select the **Step Transitions** tab and select the **New Step** button



The Process Wizard Editor creates a new step with the default name 3A (<New Step>).

- 3 Select the new step in the graphic view and rename it to Local Email Solution

We specify the condition for the local email solution step, as described in [Specifying the Condition for the Local Email Solution Step on page 73](#).

### SPECIFYING THE CONDITION FOR THE LOCAL EMAIL SOLUTION STEP

To define the transition from the email location step, we specify the condition using the Conditions for Selected Transition table of the Step Transitions tab.

We use the Field Name and Value Names that we defined in [Defining User Options for the Email Location Step on page 72](#) to specify that the transition to the local email solution step occurs when users select the local email option from the email location step.

- 1 Select the **Field Name** column of the Conditions for Selected Transition table, then select the field **2A.email\_location** from the drop-down list

Step Fields   Step Transitions   Step Preview				
Select	Go To Step			Actions
	3A (Local Email Solution)			Delete
				New Step
Conditions for Selected Transition				
Select	Field Name	Op	Value	Actions
	1A.message_type			
	2A.email_location			

- 2 Select the **Value** column, then select the value **local** from the drop-down list

Conditions for Selected Transition				
Select	Field Name	Op	Value	Actions
	2A.email_location	=	local	Delete
			local	
			remote	

**Note:** The Process Wizard Editor selects the Operator (Op) = by default. See Specifying Transition Operators for more information.

We can now define the html field for the local email solution step, as described in [Defining the Text Field for the Local Email Solution Step on page 74](#).

## DEFINING THE TEXT FIELD FOR THE LOCAL EMAIL SOLUTION STEP

The local email solution step contains only one field, an html field that explains how to retrieve email messages from the user's home computer.

The Step Fields table contains a default entry for an html field, which we edit for our process.

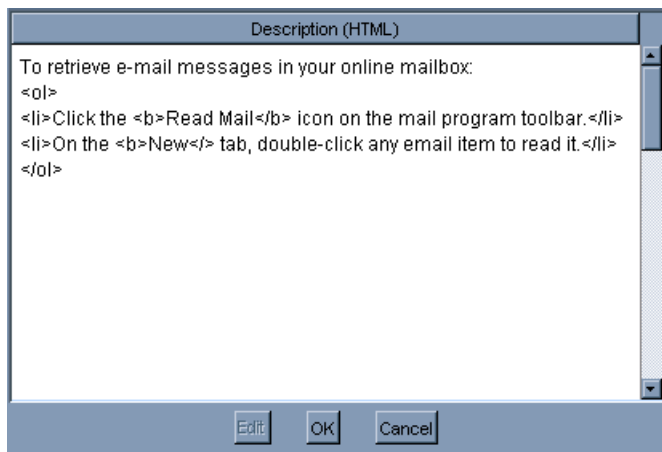
To edit the default entry:

- 1 Select the default html entry

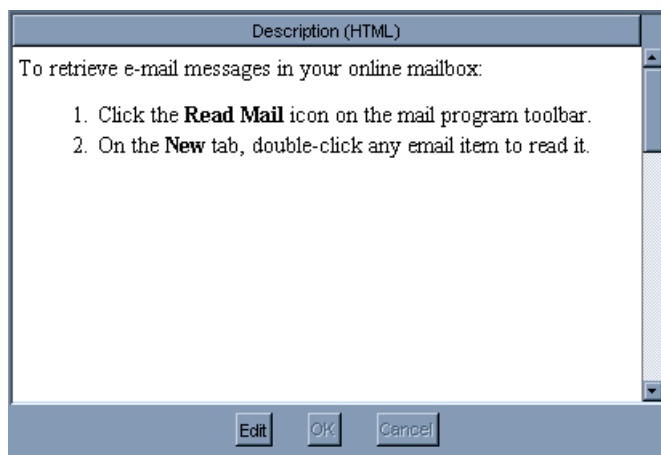
Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete

- 2 Select the **Edit** button below the Description field, and enter the solution text:

**Note:** The following provides an example of simple html markup.

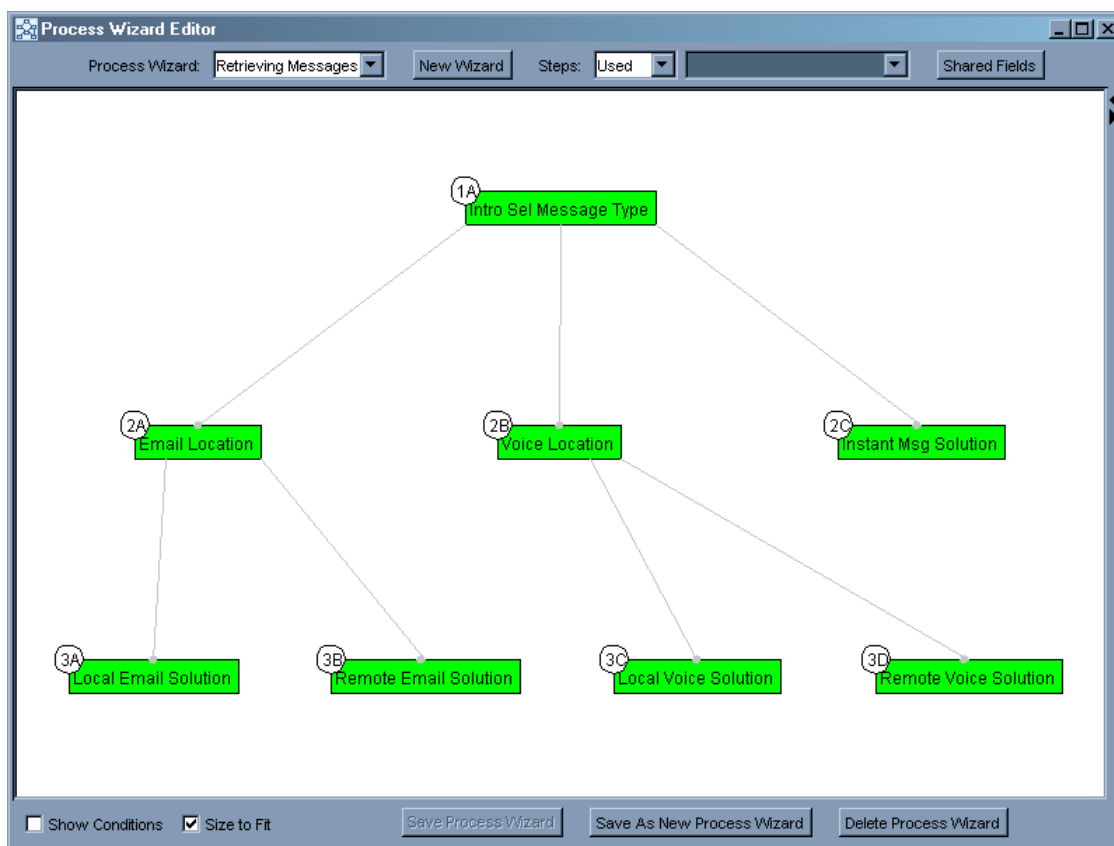


When you save the step, the Process Wizard Editor formats the html markup in the field:



## Tutorial Review

You have now completed the Process Wizard tutorial; note that we did not provide a detailed description of creating each sibling step in each row; however, the following illustration provides a view of a completed Process Wizard based on the scenario described in [Scenario: Creating a Process to Assist Message Retrieval](#) on page 57 for your reference:



## Creating Process Wizards

You create process wizards for your application using the Process Wizard Editor application in the Oracle Knowledge Language Workbench to:

- Create a new process wizard that includes the step, field, condition, and transition definitions, as described in [Creating New Process Wizards on page 77](#).
- Define the individual steps within the process wizard, as described in [Creating Process Steps on page 79](#).
- Define the fields, conditions, and transitions within the steps, as described in [Defining Fields on page 84](#), [Defining Conditions on page 93](#), and [Defining Transitions on page 94](#).

## Creating New Process Wizards

You can create new process wizards:

- Using the New Wizard button to create an empty process wizard, as described in [Defining a New Process Wizard on page 77](#)
- Using the Save As New Process Wizard function to create a copy of an existing wizard, as described in [Copying an Existing Process Wizard on page 78](#)

**Note:** The Process Wizard Editor assigns a unique internal ID to each process wizard in the form: pw\_<sequence\_number>.

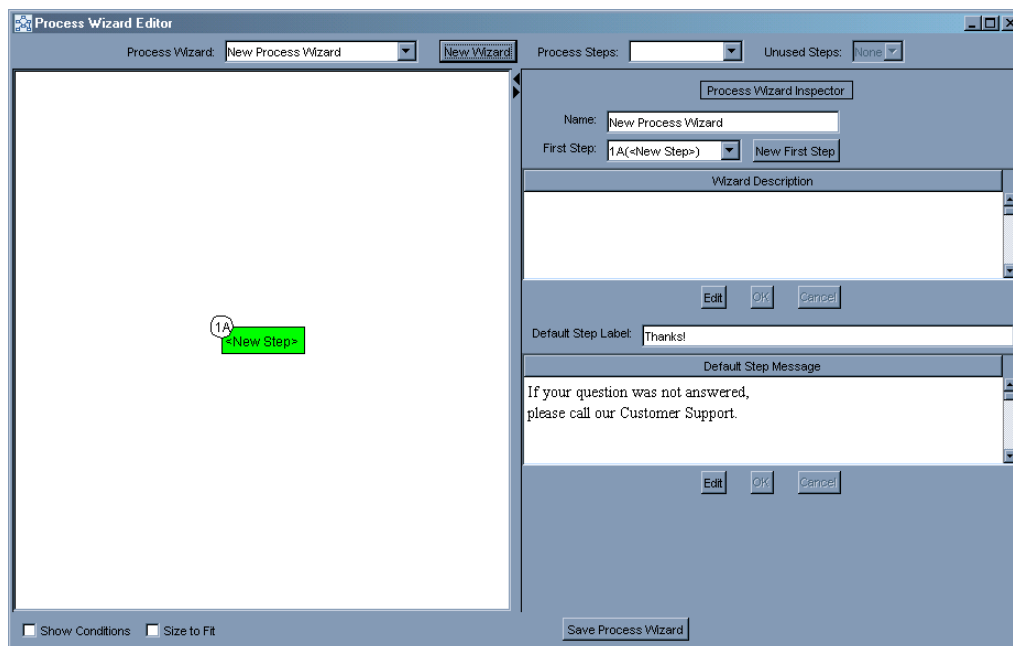
## Defining a New Process Wizard

To define a new process wizard:

- 1 Select the **New Wizard** button on the Process Wizard Editor screen:



The Process Wizard Editor creates a new process wizard named New Process Wizard containing only the default first step, 1A(<New Step>).



- 2 Begin defining steps for your process wizard by overwriting the default step parameters and defining additional steps, as described in [Creating Process Steps on page 79](#).

## Copying an Existing Process Wizard

You can copy an existing process wizard for use as a template to create additional process wizards, or for any other purpose, using the Save As New Process Wizard function in the Process Wizard Inspector.

The Save As New Process Wizard button is enabled whenever a previously saved wizard is selected.

To copy an existing process wizard:

- 1 Select the process wizard in the Process Wizard Editor, then select the **Save As New Process Wizard** button



The Process Wizard Editor displays the New Process Wizard Name dialog:



- 2 Enter the desired name for the new process wizard and select **OK**

The Process Wizard Editor displays the new process wizard.



## Creating Process Steps

You can create process steps using the New Step function in the Step Transitions tab of the step inspector.

When you create a new step, the Process Wizard Editor:

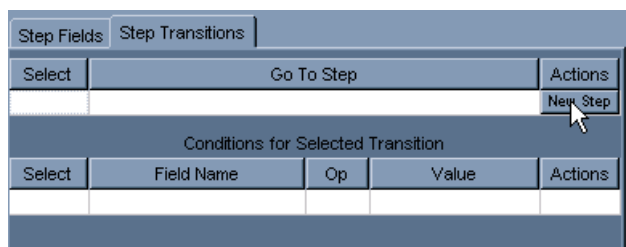
- Automatically assigns the new step to the level below the step from which it was created
- Labels the step according to its level, and the order in which it was created

For example, the first step that you create from the default step is labeled 2A (<New Step>). The next step that you create at the second level is labeled 2B (<New Step>), and the first step that you create at the third level (child of 2A or 2B) is labeled 3A (<New Step>).

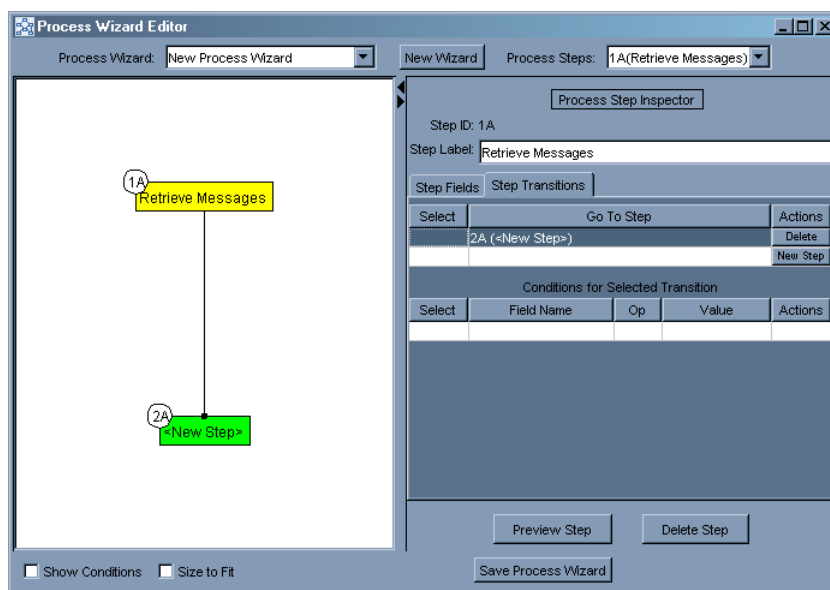
**Note:** You can also create a new top level step, as described in [Creating a New First Step on page 80](#) or [Designating a New First Step on page 81](#).

To create a process step:

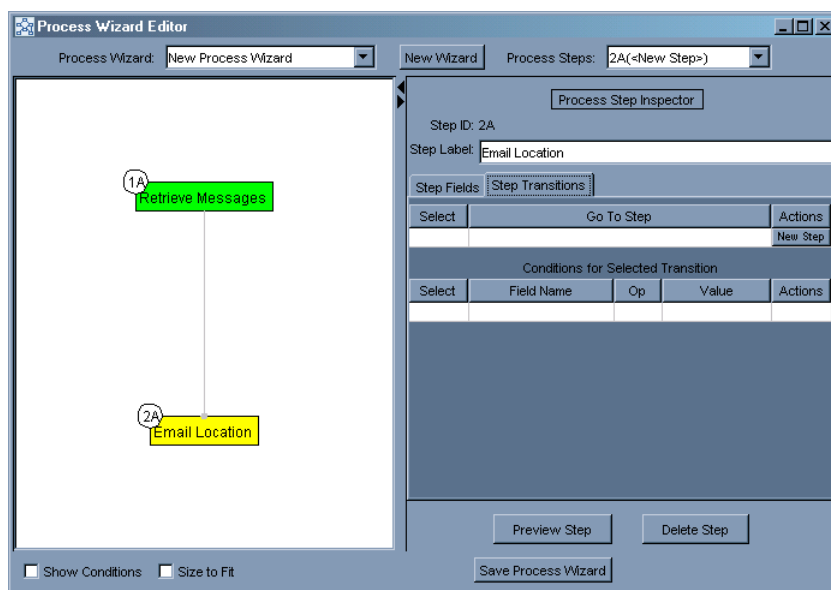
- 1 Select an existing step to be the parent of the new step
- 2 Select the **Step Transitions** tab, then select the **New Step** button



The Process Wizard Editor displays the new step, with the default label <New Step>:



- 3 To change the step label, select the new step and edit the **Step Label** field



You can define fields, conditions, and transitions for the new step, as described in [Defining Fields on page 84](#), [Defining Conditions on page 93](#), and [Defining Transitions on page 94](#).

## Creating a New First Step

You can create a new initial step for your process wizard using the New First Step function.

**Note:** When you create a new process wizard, the Process Wizard Editor automatically creates new step in the initial position of the process, which you then modify, as described in [Defining Fields on page 84](#), [Defining Conditions on page 93](#), and [Defining Transitions on page 94](#).

The New First Step function is available only from the Process Wizard Inspector.

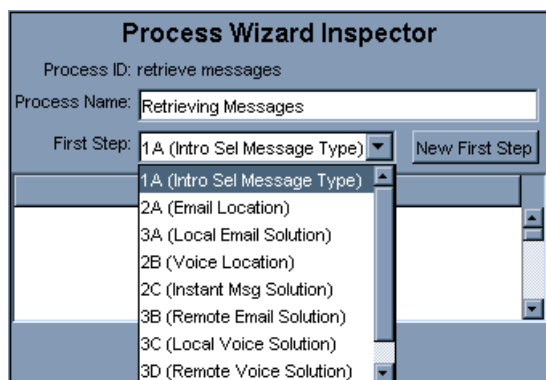
To create a new first step, select the **New First Step** button



The Process Wizard Editor inserts a new first step above the current first step to the process, and re-labels the lower level process steps.

## Designating a New First Step

You can designate an existing step as the new first step in a process wizard using the First Step menu of the Process Wizard Inspector.



The First Step menu lists all of the steps in the process wizard.

**Important!** You can specify any step as the first step in the process; however, the steps, conditions, and transitions above the new first step are excluded from the process, and added to the unused step list.

To assign an existing step as the first step in the process, select the desired step from the **First Step** drop-down menu

The Process Wizard Editor re-structures the process wizard, and adds any excluded steps to the unused steps list.

## Deleting a Process Step

You can delete steps from a process using:

- The Delete function of the step inspector
- The Delete Step function within the graphic view

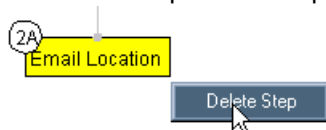
**Important!** When you delete a step that is the only source of transitions to a subsequent step, the transition is deleted and the subsequent step becomes an unused step, as described in [Unused Steps on page 82](#).

To delete a process step, select the **Delete Step** function of the Step Transitions tab within the Process Wizard Editor:



or

Right-click the step within the process diagram area and select **Delete Step**



The Process Wizard Editor deletes the selected step, and re-structures the process as necessary.

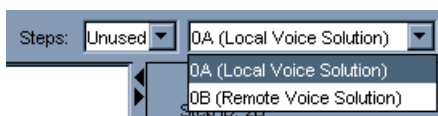
## Unused Steps

The Process Wizard Editor allows you to define and save steps that are not used within the process. The Process Wizard Editor classifies any step that is not accessible from the initial step by a series of transitions as an unused step. You can display unused steps using the Steps menu.

**Note:** The Process Wizard Editor also displays unused steps as candidates for transitions, as described in [Defining Transitions on page 94](#).

To display an unused step:

- 1 Select **Unused** from the Process Wizard Editor Steps menu:



The Process Wizard Editor lists the unused steps.

- 2 Select an unused step to display its details in the Process Step Inspector

## Previewing Steps

You can view the steps within a process wizard using:

- The Step Preview tab of the Process Step Inspector, as described in [Previewing Steps within the Process Step Inspector on page 82](#)
- The Preview Step in Browser function, as described in [Previewing Steps within a Browser on page 83](#)

## Previewing Steps within the Process Step Inspector

You can view the basic content and format of a step using the Step Preview tab of the Process Step Inspector.

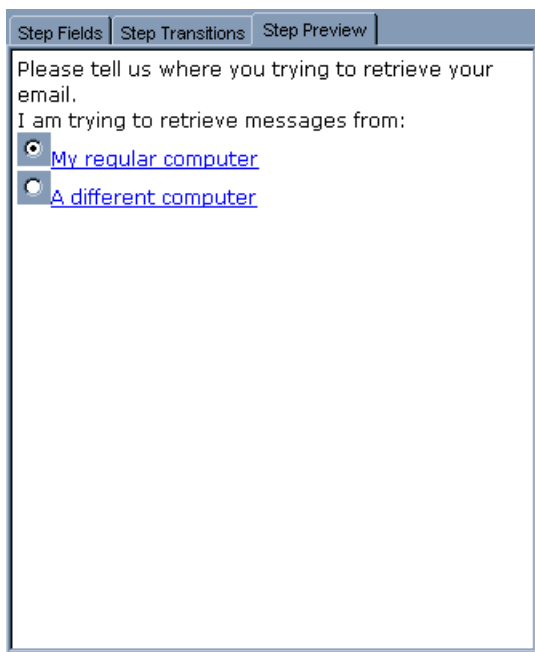
The Step Preview tab displays basic formatting for selected steps; however, the HTML rendering is not as complete as when viewed using the browser preview function.

This preview mode is useful for quickly reviewing a process by selecting the Step Preview tab, then selecting multiple steps in sequence from the graphic view area.

To display a preview of a selected step, select the **Step Preview** tab



The Process Wizard Editor displays the formatted contents of the current step:



You can preview other steps in the process by selecting them in the graphic view.

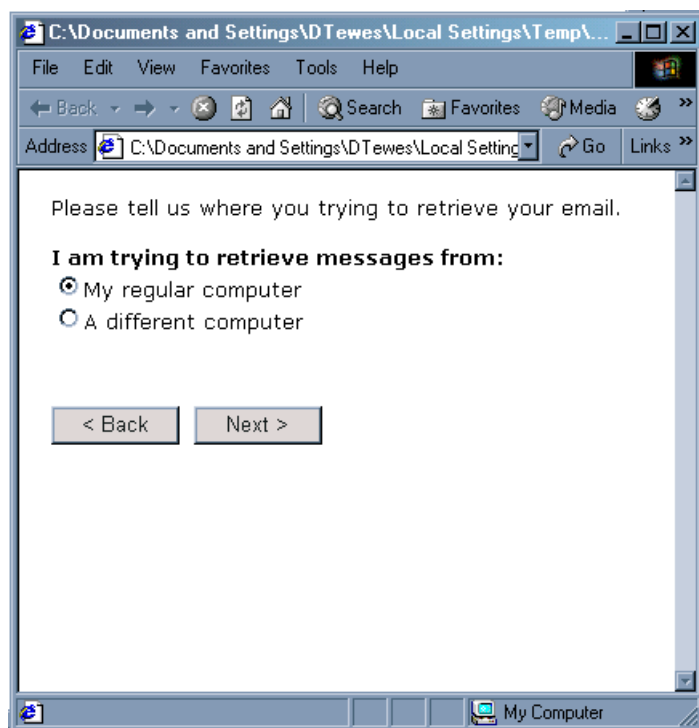
## Previewing Steps within a Browser

You can view detailed formatting for a single step using the Preview Step in Browser function of the Step Inspector. The Preview Step in Browser function displays the current step in a separate browser window using basic formatting.

To display a preview of a selected step, select **Preview Step in Browser**



The Process Wizard Editor displays the current step in a separate browser window:



**Note:** The **< Back** and **Next >** buttons are not functional in preview mode.

## Defining Fields

You can define fields within process steps using the Step Fields tab within the Process Wizard Editor. Step fields define the components of the step, such as informational text and user options that are displayed in the User Interface.

The User Interface displays step fields in the order in which they are defined in the Step Fields table. The bottom row of this table is always empty to accommodate a new option. You can change the order of fields within a step by selecting the step to be moved using a Mouse Down in its **Select** column and dragging it to the desired location then dropping it with a Mouse Up.

**Note:** When you create a new field, the Process Wizard Editor creates a default html field.

To define a new field:

- 1 Select the **Type** cell in the last row of the Step Fields area, then select the desired Type for the new field from the drop-down menu

**Note:** See [Specifying Field Types on page 86](#) for more information on the types of fields you can define within steps.

The Process Wizard Editor creates the selected field type, for example:

- 2 Specify the following additional parameters for the field:

Parameter	Description
<b>Name (For Conditions)</b>	Use this parameter to specify a unique ID that you used to refer to the field when specifying Conditions and Transitions, and any type-specific attributes, as described in <a href="#">Defining Conditions on page 93</a> .
<b>Description</b>	Use this parameter to specify: <ul style="list-style-type: none"> <li>• formatted text for html fields</li> <li>• formatted introductory text for user option fields (checkbox, radio, select, text)</li> </ul>
<b>Additional Attributes</b>	See <a href="#">Defining Conditions on page 93</a> and <a href="#">Defining Transitions on page 94</a> for more information on the attributes associated with each field type.

You can delete or change the order of fields within steps using the Delete and up and down arrows in the Step Fields tab:

## Specifying Field Types

You specify the Type for a field to determine whether the field is informational (html), or whether it presents user options for use in setting conditions for subsequent step transitions.

The following table describes the available field types:

Type	Description
<b>checkbox</b>	<p>Use this field to define a set of options where users can select one or more options from the set by clicking in empty checkboxes. To specify a checkbox field, enter:</p> <ul style="list-style-type: none"> <li>• An optional heading or introductory text for the checkbox options in the Description field</li> <li>• Labels for each checkbox option in the Conditions Value field</li> <li>• A brief Value Name for use in specifying transitions</li> <li>• An optional default selection from among the options</li> </ul> <p>See <a href="#">Defining Conditions on page 93</a> for more information.</p> <div data-bbox="516 814 928 1087"> <p><b>Sample Checkbox Field:</b></p> <p><input checked="" type="checkbox"/> item A</p> <p><input type="checkbox"/> item B</p> <p><input type="checkbox"/> item C</p> <p>Back Finish</p> </div>
<b>html</b>	<p>Use this field to define an informational field. You specify an html field by entering any valid html in the Description field of the Step Inspector. The User Interface displays the formatted html, for example:</p> <div data-bbox="490 1234 854 1417"> <p>Sample HTML Field...</p> <p>Back Finish</p> </div>



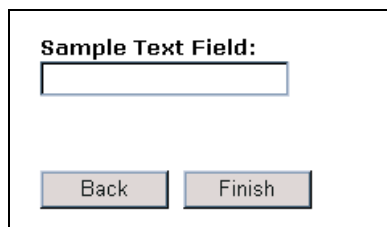
<b>radio</b>	<p>Use this field to define a set of options where users can select only one option from a list of radio buttons. To specify a radio field, enter:</p> <ul style="list-style-type: none"> <li>• An optional heading or introductory text for the radio options in the Description field</li> <li>• Labels for each radio option in the Conditions Value field</li> <li>• A brief Value Name for use in specifying transitions</li> <li>• An optional default selection from among the options</li> </ul> <p>See <a href="#">Defining Conditions on page 93</a> for more information.</p> <div data-bbox="548 478 956 758"> <p><b>Sample Radio Button Field:</b></p> <p><input checked="" type="radio"/> item A  <input type="radio"/> item B  <input type="radio"/> item C</p> <p>Back Finish</p> </div>
<b>search</b>	<p>Use this field to launch an Oracle Knowledge search based on the user responses to the fields within the process wizard, as described in <a href="#">Specifying a Search Action within a Step Field on page 89</a>.</p> <div data-bbox="495 894 1320 1123"> <p><b>Sample Search Field:</b></p> <p><input type="text"/> <input type="button" value="Search"/></p> <p>Back Finish</p> </div> <p>The <b>search</b> option creates a search button on the Process Wizard UI which subsequently invokes the search.</p>
<b>select</b>	<p>Use this field to define a set of options where users can select only one option from the set. To specify a select field, enter:</p> <ul style="list-style-type: none"> <li>• An optional heading or introductory text for the select listbox in the Description field</li> <li>• Labels for each selection option in the Conditions Value field</li> <li>• A brief Value Name for use in specifying transitions</li> <li>• An optional default selection from among the selection options</li> </ul> <p>See <a href="#">Defining Conditions on page 93</a> for more information.</p> <div data-bbox="527 1514 924 1738"> <p><b>Sample Select List Field:</b></p> <p>item A  item A  item B  item C</p> <p>Back Finish</p> </div>

**text**

Use this field to define an area in which users can enter free-form text.  
To specify a text field, enter:

- An optional heading or introductory text for the text field in the Description field
- A label for the text field in the Conditions Value field
- A brief Value Name for use in specifying transitions
- An optional default selection from among the options

See [Defining Conditions on page 93](#) for more information.

A screenshot of a 'Sample Text Field' dialog box. It features a title bar at the top with the text 'Sample Text Field:'. Below the title bar is a single-line text input field. At the bottom of the dialog box, there are two buttons: 'Back' on the left and 'Finish' on the right.

**Note:** This is required for search steps. See [search](#), above.

**Note:** The Type of CheckBox, Radio, and Select fields can be changed to either of the other two types with Options without changing the options.

## Specifying a Search Action within a Step Field

To define a search action within a step field, you associate the search field with one or more context variables. In the **Field to Variable Map for selected Field** section at the bottom of the Process Step Inspector, enter the Field Name of each value to be attached to the original question when it is resubmitted and the Variable Name it is to be assigned to.

**Note:** You must have context variables defined within your application.

The application resubmits the user's original question, that triggered the display of the Process Wizard, with the context variables set with the values obtained by the previous Process Wizard steps.

**Note:** It is important to add a Business Condition to the Rule that triggers the Process Wizard Answer to not be true if any of the Context Variables, associated with this Process Wizard's Search fields are set; otherwise the Process Wizard answer displays again.

When you specify a search field type, the Process Wizard User Interface replaces the Next or Finish button with a button labeled Search.

**Important!** You can specify the search action only in the steps which have no transitions to other steps.

When an end user invokes search from the Process Wizard User Interface, the application submits a request containing:

- The original question
- The context variables and their assigned values, specified for the Search Field invoked.

## Creating Shared Fields

You can create, view, and modify shared fields for use within multiple Process Wizards. You can use shared fields:

- As standard fields shared among multiple process wizards
- As templates for creating similar fields

Shared fields enable you to create and manage a collection of commonly used fields and templates for use within process steps.

For example, you could define a global **Yes/No** radio button field, which you could use:

- In its basic form wherever a simple Yes or No response is required
- As a template for defining Yes/No options that require additional information or qualification

You can also create shared fields containing a standard message and set of options for choices that are presented consistently in multiple processes.

For example, you could create select fields for products, operating systems, or colors.

You define shared fields in the same way that you define basic step fields.

**Note:** You cannot define a shared field using the search field type; search field definitions require internal mapping that is specific to the Process Wizard in which it is used.

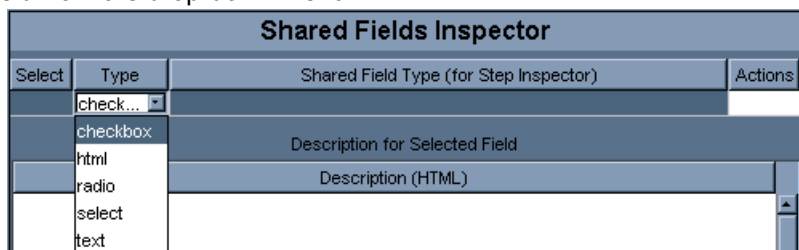
To define a shared field:

- 1 Select the **Shared Fields** button near the top of the Process Wizard Editor



The Process Wizard Editor displays the Shared Fields Inspector.

- 2 Select the **Type** cell in the last row of the Shared Fields area, then select the desired Type for the new field from the drop-down menu



**Note:** See [Specifying Field Types on page 86](#) for more information on the types of fields you can define within shared steps.

The Process Wizard Editor creates the selected field type, and supplies a default field name, for example:



- 3 Specify the following parameters for the shared field:

Parameter	Description
<b>Shared Field Type (For Step Inspector)</b>	Use this parameter to specify a type name for use in the Step Inspector. The Step Inspector uses the value of this field to display this shared field as an option in the Type menu, as described in <a href="#">Adding a Shared Field to a Process Step on page 91</a> .
<b>Description</b>	Use this parameter to specify: <ul style="list-style-type: none"> <li>• formatted text for html fields</li> <li>• formatted introductory text for user option fields (checkbox, radio, select, text)</li> </ul>
<b>Value Name (For Conditions)</b>	Use this field to specify a value name for each option within the set of user options defined for this field.
<b>Value</b>	Use this field to specify the text to display within the User Interface for each condition name within the set of user options defined for this field.
<b>Default</b>	Use this field to specify whether this option should be pre-selected by default.

Options for Selected Field			
Value Name (for Conditions)	Value	Default	Actions
		<input type="checkbox"/>	

- 4 Save the shared field by selecting the **Save Shared Field** button.

The Process Wizard Editor saves the new shared field.

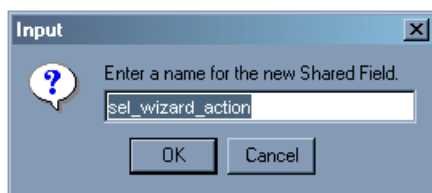
## Creating a New Shared Field from an Existing Shared Field

You can create a new shared field from an existing shared field by modifying the existing field. To save the modified field as a new shared field:

- 1 Select the **Save as New Shared Field** button:



The Process Wizard Editor displays a dialog that prompts you to specify a name for the new shared field.



- 2 Enter the new name and select **OK**

The Process Wizard Editor adds the new shared field to the list in the Shared Fields Inspector.

## Adding a Shared Field to a Process Step

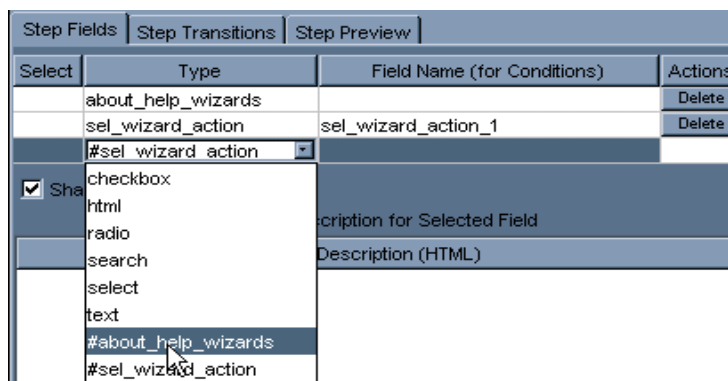
You can add shared fields to a process step by selecting the shared field from the list of fields that display in the Type column of the step fields tab. The Process Wizard Editor indicates shared fields by prefixing the field name with the # character, for example:

#about\_help\_wizards

To add a shared field to a process step:

- 1 Select the **Step Fields** tab for desired step, then select the **Type** field

The Process Wizard Editor displays the available shared fields in the list of field types:



- 2 Select the desired shared field

The Process Wizard Editor adds the shared field to the selected step.

## Using a Shared Field as a Template

If you want to use a Shared Field as a template, such as wanting to retain all of the options of the changed field, but change its Description, you can add the shared field to your Step, as shown in Adding a Shared Field to a Process Step, above, then clear the **Shared Field** checkbox, while the field is still selected. This changes the reference to the Shared Field to a copy of the Shared Field and reverts its **Type** to the base Type of the Shared Field. You can then modify the field, as needed, without affecting the Shared Field. Unlike having a reference to the Shared Field in your step, any future changes to the Shared Field are not reflected in the copy.

## Creating a Shared Field from a non-Shared Field

Often, the first time you create a field in a step, you don't foresee needing a field with the same options, description, etc. again. Then later, you realize that you need to create a field just like one you already created.

- 1 Find the original field that you would like to be a Shared Field, even if it is in another Process Wizard.
- 2 Select the field.
- 3 Select the **Shared Field** checkbox, in the lower-left corner of the Step Fields tab.
- 4 This prompts you to enter a **Type** name for the Shared Field,
- 5 Save the original field as a Shared Field.
- 6 Replace the original field with a reference to the Shared Field.
- 7 Save the Process Wizard with the original field, if different than the one you were working on, then go back and create the new field by selecting the new Shared Field Type.

## Defining Conditions

Conditions are parameters that you define within step fields that offer multiple user options. You use conditions to specify the transition to the next step in the process for each user option. For example, a step field that asks a user to select a `processor_type` might specify a transition to a different step for each possible response.

You specify conditions for the following types of fields:

- Checkbox
- Radio
- Select

You define conditions for the user option fields within process steps by specifying:

- The Field Name (For Conditions) field, as described in [Specifying Condition Field Names on page 93](#)
- An operation and a value in the Options for Selected Field table, as described in [Specifying Condition Values on page 93](#)

## Specifying Condition Field Names

To specify a field name for a condition:

- 1 Select the type of condition (checkbox, radio, or select)

The Process Wizard Editor automatically enters a default field name corresponding to the selected field type:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete
	radio	radio_1	Delete
Description for Selected Field			

- 2 Rename the field name, for example:

Step Fields   Step Transitions   Step Preview			
Select	Type	Field Name (for Conditions)	Actions
	html		Delete
	radio	ProcessorType	Delete
Description for Selected Field			

- 3 Specify condition values for each user option in the field, as described in [Specifying Condition Values on page 93](#)

## Specifying Condition Values

You specify condition values by specifying:

- The condition value
- The display value

- Whether an option is selected by default when the step is displayed

in the Options for Selected Field table.

To Specify condition values, enter appropriate values in the fields, for example:

Options for Selected Field			
Value Name (for Conditio...	Value	Default	Actions
NoPentium	Not a Pentium	<input type="checkbox"/>	Delete

Parameter	Description
<b>Value Name (For Conditions)</b>	Use this field to specify a value name for each option within the set of user options defined for this field.
<b>Value</b>	Use this field to specify the text to display within the User Interface for each condition name within the set of user options defined for this field.
<b>Default</b>	Use this field to specify whether this option should be pre-selected by default.

## Defining Transitions

Transitions are the mechanisms that determine the next step to display in the process based on a user-selected condition. You specify transitions for the condition values (user options) that you define within a step.

**Note:** If you do not specify a transition for a condition, the Process Wizard displays the default step when a user selects that condition.

You specify transitions using the Step Transitions tab of the Process Wizard Editor.

To specify a transition for a step:

- 1 Select the **Step Transitions** tab

Step Fields	Step Transitions	Step Preview			
Select	Go To Step				Actions
					New Step
Conditions for Selected Transition					
Select	Field Name	Op	Value	Actions	

- 2 Specify the following transition fields:

Field	Description
<b>Go to Step</b>	Use this field to select or create the step to which you want to transition.
<b>Actions</b>	Use this field to create a new process step for the transition, or to delete a transition.
<b>Conditions for Selected Transition</b>	This table displays the following fields to specify conditions:



<b>Field Name</b>	Use this field to specify the condition that applies to the currently selected transition step (the Go to Step field).
<b>Op (Operator)</b>	Use this field to specify an operator to evaluate the condition.
<b>Value</b>	Use this field to specify the value that you want to apply to the currently selected transition step.

## Creating Dictionary Rules to Invoke Process Wizards

You configure a process wizard for use by the application by creating a rule that invokes the process wizard as a response (action) to the appropriate user request and context (condition).

To create process wizard rule:

- 1 Create a rule and define the rule's condition so that it matches the desired request(s) using the Dictionary Manager, as described in *Intelligent Search Language Tuning Guide, Chapter 2: Dictionary Manager, Working with Rules on page 70* .  
If the Process Wizard has any Search fields, you must add a Business Condition that keeps the rule from firing if any of the Context Variables set by the Search fields are set; otherwise, when the question is resubmitted, with the added Context Variable settings, the Process Wizard is displayed again.
- 2 Define the rule's action to invoke the desired process wizard, as described in *Assigning the Process Wizard Action to a Rule on page 95*

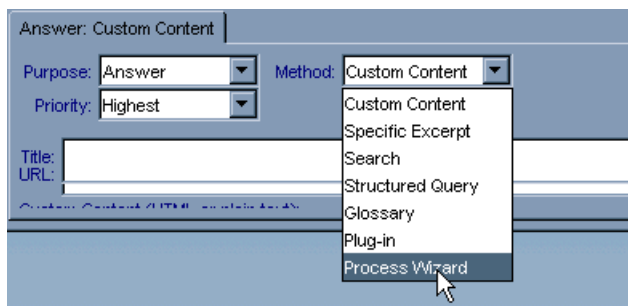
## Assigning the Process Wizard Action to a Rule

You assign the process wizard to a rule by specifying the rule's action using the Answer purpose and the Process Wizard answer method.

**Note:** The Answer purpose is the only supported purpose for displaying process wizards in the User Interface.

To specify the process wizard answer method:

- 1 Select the **Answer** purpose from the Purpose drop-down menu
- 2 Select **Process Wizard** from the Methods drop-down menu on the rule's Actions tab



The Dictionary Manager displays a drop-down menu of the process wizards currently defined in the Dictionary.

- 3 Select the desired process wizard from the drop-down menu

The rule is available to the application when you commit your changes to the Dictionary.

## Creating a Process Wizard Intent Response

As an alternative to creating a Rule with a Process Wizard Action, a Process Wizard can be displayed when an Intent triggers by creating a Process Wizard Intent Response.

To create a Process Wizard Intent Response:

- 1 Select either the Intent Response Icon on the Dictionary Manager Tool Bar or the File/New Intent Response menu item.
- 2 Select **yes** when prompted *Do you want to create a Process Wizard Response*.
- 3 When the Process Wizard Intent Response Editor displays:
  - a Enter a name for the Intent Response.
  - b (Optional) Enter a Description and any Notes.
  - c Select the Domain for the new Intent Response.
  - d Select the **User Profile** option you want to use or leave the default All Users setting.
  - e Select a language for the Intent Response by selecting the desired language and clicking **Add Selected Language to Response**.

When you select the Language for the Response, a tab appears for the selected language, with a drop-down menu of available Process Wizards.

- 4 Select the Process Wizard you want to display.
- 5 Assign the Intent Response to one or more Intents by selecting the names of the Intents, one at a time, in the **Intents Assigned to the Response** table at the bottom of the Intent Response Editor.

## The Process Wizard User Interface

The Process Wizard User Interface is a set of specific pages designed for use with Process Wizards. When users select an Process Wizard answer from the standard answer page, the User Interface invokes the Process Wizard User Interface pages to display the selected Process Wizard.

**Note:** The Process Wizard User Interface is automatically configured for use within the standard User Interface.

The Process Wizard User Interface consists of the following major elements:

- The Process Wizard answer, which displays on the answer page, as described in [The Process Wizard Answer on page 97](#)
- The step display area, which contains the steps defined for the process, as well as the navigation buttons (Back, Next, Finish), as described in [The Step Display Area on page 97](#)
- The process summary column, which displays information about the previous steps that the user has taken to progress through the wizard, as described in [The Step Display Area on page 97](#)

## The Process Wizard Answer

When an end-user submits a request to the application that matches a process wizard rule, the User Interface displays a special Process Wizard answer in the standard answer area, for example:

### Find Answers

Type a question or describe what you are looking for below



Example: "Does Product-X have Feature-Y?"

[Tips](#)

### Answers



#### [Help Retrieving Messages](#)

Having problems retrieving messages? Let us guide you through it.

If users select the link in the process wizard answer, the User Interface displays the initial step of the process wizard.

## The Step Display Area

When users select a Process Wizard answer, the application displays the initial step in the Process Wizard User Interface step display area.

The Process Wizard User Interface displays a summary of the user's previous responses to the left of the step display area. The summary displays below the heading **Your Responses**.

Each response is displayed as a link that navigates back to the process step.

<a href="#">» Start over</a>	<h3>Help Retrieving Messages</h3> <p>Having problems retrieving messages? Let us guide you through it.</p>
<h3>Your Responses</h3> <p>What kind of message are you trying to retrieve? <b>Email Message</b></p> <p>From where are you trying to retrieve your messages? <b>From your home computer</b></p>	<p>To read e-mail in your online mailbox</p> <ol style="list-style-type: none"> <li>1. Click the Read icon on the toolbar.</li> <li>2. On the New tab, double-click the first e-mail item to read it.</li> </ol> <div> <input type="button" value=" &lt; Back"/> <input type="button" value=" Finish"/> </div>

## Modifying the Process Wizard User Interface

You can modify Process Wizard User Interface elements to suit the needs of your application by editing the Process Wizard User Interface files, located in one of the following locations:

`<Oracle_Knowledge_root>/Search/inquire/webapps/<subdirectory>`

where:

`<subdirectory>` is one of the following:

- `css`
- `js`
- `xsl/search`

CSS Files	Description
<code>qna_wizard_style.css</code>	This is the style sheet that specifies the style and formatting for the elements that are specific to the Process Wizard User Interface and are not part of the standard search.
Java Script Files	Description
<code>qna_wizard.js</code>	This is a JavaScript library that contains Process Wizard User Interface-specific functionality.
XSL Files	Description
<code>wizard.xsl</code>	This is the main Process Wizard User Interface file that contains the basic page definition (similar to the <code>ui_search*.xsl</code> files) and utilities for the wizard pages.
<code>wizard_fields.xsl</code>	This file contains all of the templates used to render any defined wizard fields such as radio buttons, text boxes, select boxes, HTML areas, etc. on the UI.
<code>wizard_history.xsl</code>	This file contains the templates for displaying the user's choice history as well as the support templates for any actions generated by the history information such as links back to previous pages.

**Note:** The file `/webapps/xsl/process_wizard/step.xsl` is used only for previewing steps in the Process Wizard Editor, and is not used in the Process Wizard User Interface.

# Personalized Navigation Setup

You can implement the Oracle Knowledge Personalized Navigation feature to categorize application content and display a hierarchy of the results, such as products and models, services, and prices, in the User Interface. When Personalized Navigation is implemented, users can select among content categories and other types of classification to restrict the results to the selected areas of interest.

You can also integrate Personalized Navigation with Oracle Knowledge question answering to automatically generate answers to natural language questions based on the categories of information referenced in users' questions. For example, users can get results related to a specific price range either by selecting the appropriate price facet or by referring to a price in their question, once a Rule is setup to automatically restrict the answers to the desired facet.

Personalized Navigation provides end users with the choice of displaying application content by selecting classification links within the navigation area of the question page. For example, users can select from content classifications displayed in the User Interface:

Find Answers

>> Start over

Filter Answers By:

Author

☐ [Abbey, Lynn](#) (2)
 ☐ [Acres, Mark](#) (2)
 ☐ [Adams, Douglas](#) (15)
 ☐ [Adams, Robert](#) (2)
 ☐ [Affabee, Eric](#) (1)
 [more...](#)

Year of Publication [x]

☐ [1970-1979](#) (328)
 ☒ [1980-1989](#) (1916)
 ☐ [1980](#) (134)
 ☐ [1981](#) (155)
 ☐ [1982](#) (247)
 ☐ [1983](#) (279)
 ☐ [1984](#) (155)
 ☐ [1985](#) (191)

Example: "Does Product-X have Feature-Y?"

[Tips](#)

Answers

Available Books

Cover	Author	Title	Price	Year
	Isaac Asimov	100 Great Science Fiction Short Short Stories	\$3.95	1985
	Arthur Charles Clarke	2001: A Space Odssey	\$4.95	1982
	Arthur Charles Clarke	2010	\$3.95	1987

[View entire table \(100 rows\)](#)

The links in the navigation area (left column) correspond to the content categories (also called facets). Users can select from the available categories to locate a particular item of interest.

Personalized Navigation can also refer to Dictionary rules so that a user question such as "What books do you have between \$10 and \$20?" automatically selects content categories.

Personalized Navigation includes built-in data classification tools and methods to:

- **Define any number of categories and subcategories for classification.**

Oracle Knowledge provides a variety of built-in classification methods to classify structured and unstructured data, including methods to access database fields, site graph, meta-tags, and concept relations.

- **Classify application data at multiple levels.**

You can classify application content at the sentence level, section level, or any other portion of a document, and create Dictionary rules to restrict results based on classification during question processing.

- **Leverage Dictionary concept definitions to categorize data.**

You can use the concept information stored in the Dictionary to create taxonomies, and schedule taxonomy updates so that any changes to the ontology or content are automatically reflected in the navigation categories.

- **Classify unstructured data based on structured data.**

You can use structured data categories to access related unstructured content, so that, for example, unstructured data consisting of book reviews listed by titles can be categorized by the prices and publication years contained in a separate database.

## Personalized Navigation Presentation

Oracle Knowledge provides delivered User Interface implementation for Personalized Navigation featuring:

- Seamless integration with the standard Oracle Knowledge search implementation.

Personalized Navigation presents results based on selected content categories, enabling users to select facets that then restrict or expand the scope of their original question.

- Dictionary leveraging of Oracle Knowledge natural language processing.

Dictionary rules can select content categories when concepts are referenced in users' question, leveraging the full power of Oracle Knowledge natural language processing. For example, users get classified results from questions such as:

```
What books were published in 1992?
What books cost around $15?
What books cost between $12 and $18?
```

- Support for "upward" navigation by selection or by question processing.

Personalized Navigation supports navigation to back out of a selected category to any higher level category. More importantly, users can expand the scope of a natural language question. For example, consider a question about car models:

```
Special deals for Sputniks?
followed by selecting the Sedans facet.
```

The application expands the scope of the question to mean:

```
Special deals for (all sedans)?
```

## Personalized Navigation Implementation

You implement Personalized Navigation by creating hierarchies of information categories, called taxonomies. You define and create the various levels of your taxonomies using the Personalized Navigation application in the Oracle Knowledge Language Workbench.

The Personalized Navigation application contains tools and methods to:

- Associate your application content with the categories defined in your taxonomy.
- Generate subcategories within the taxonomy based on properties within your application content.

The basic Personalized Navigation application process uses a simple four step dialog to prompt you for the information required to build a taxonomy. You can use Personalized Navigation application advanced settings to explicitly specify content classification, as described in [Using Advanced Facet Definition Settings on page 133](#).

**Note:** You can tailor the question answering and presentation aspects of Personalized Navigation by adding custom Dictionary rules to access classification information, as described in [Referring to Facets from Dictionary Rules on page 128](#). You can also use classification in question processing without displaying navigation in the User Interface.

## The Personalized Navigation Application

You define and create taxonomies for your application content using the Personalized Navigation application in the Oracle Knowledge Language Workbench.

The Personalized Navigation application contains tools and methods to:

- Associate your application content with the categories defined in your taxonomy.
- Generate subcategories within the taxonomy based on properties within your application content.

[Personalized Navigation Application Classification Components on page 155](#) describes the components of the Personalized Navigation application.

## Personalized Navigation Application Components

The Personalized Navigation application uses the following components for classifying application data:

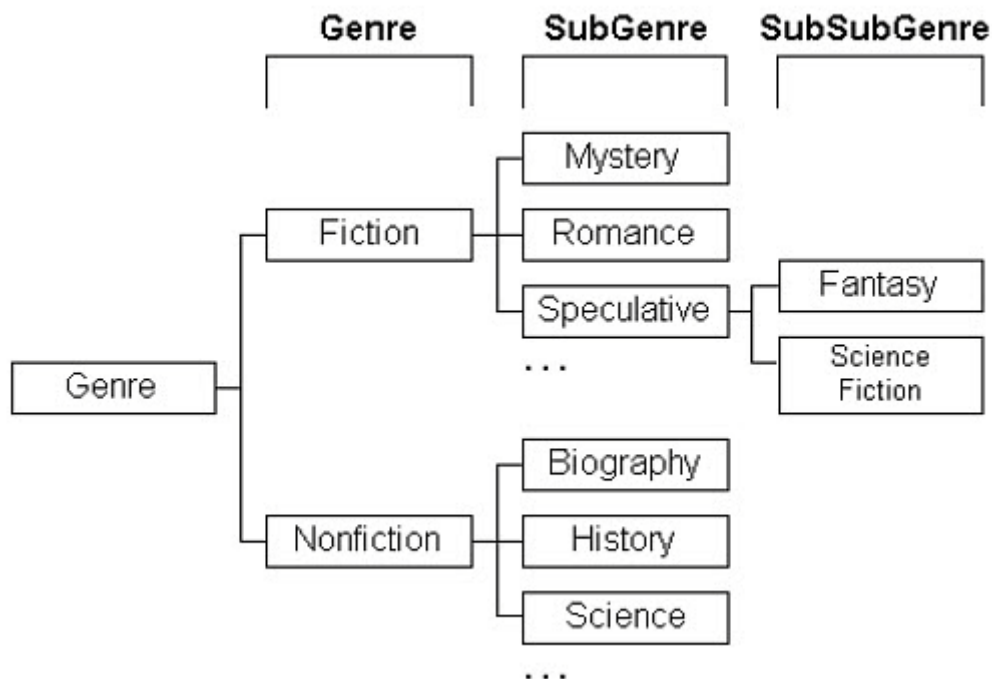
- Taxonomies – The classification schemes that organize the units of information, such as documents, sentences, and database entries, into a hierarchical structure, as described in [Taxonomies on page 101](#).
- Facets – The individual categories within a taxonomy, as described in [Facets on page 102](#).
- Classifiers – The procedures that associate facets with the relevant data within the application content, as described in [Classifiers on page 103](#).
- Taxonomy Builders – Procedures that you configure within a classifier to generate additional sub-facets during the classification process, as described in [Taxonomy Builders on page 103](#).
- Formatters – Procedures that control how facet names are presented in the User Interface, as described in [Formatters on page 103](#).
- Comparators – Modules that compare values of facet items to determine display order, as described in [Comparators on page 104](#).

In addition, there are packaged rules within the Dictionary that support Personalized Navigation functions; you create custom rules to reference specific facets, as described in [Referring to Facets from Dictionary Rules on page 128](#).

## Taxonomies

A taxonomy is a classification of objects. For Personalized Navigation the items classified in the taxonomy are items of relevant information, such as documents, sentences, and database entries, that belong to defined categories, such as product model, and price.

For any given item in the taxonomy, all objects classified in that item are implicitly classified for all parent items in the taxonomy. For example, in the taxonomy illustrated below, every Fantasy book is also a Speculative book and also a Fiction book.



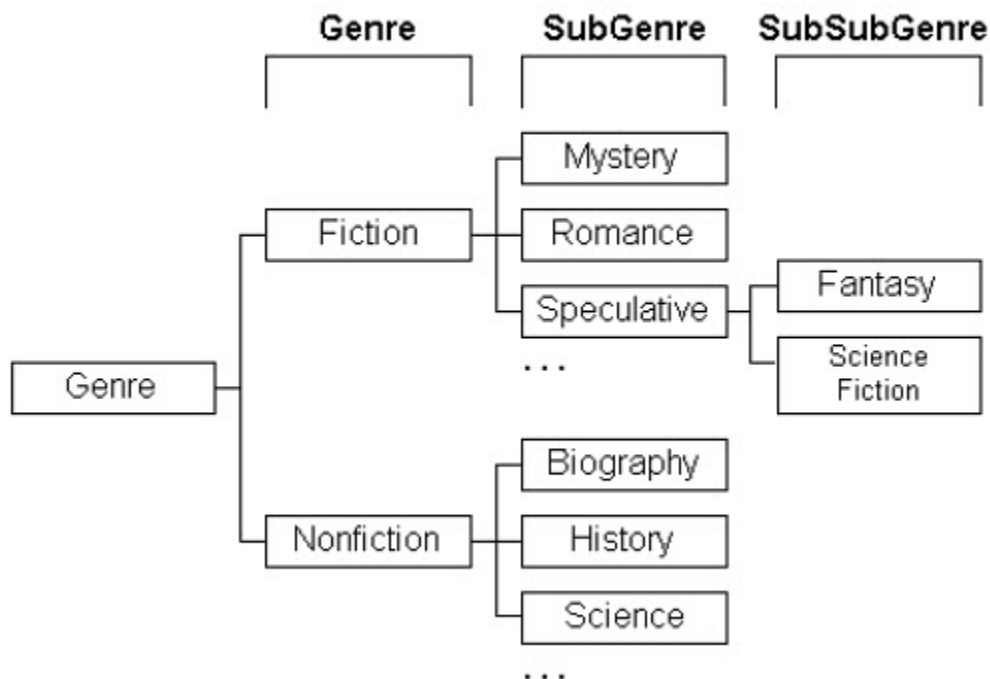
In Personalized Navigation, the categories created by the classification process (either automatically generated or manually created) are referred to as facets.

## Facets

Facets are the individual information categories within a taxonomy. You define facets that correspond to the categories of information that you want to make available to end users, either by defining them manually or by using the Personalized Navigation application classifiers.

For example, you could manually define a set of facets based on fiction book sub-genres, such as Mystery, Romance, and Speculative, and use the Personalized Navigation application classifiers to generate additional child facets based on values stored in a database.





## Classifiers

Classifiers are Personalized Navigation application procedures that you configure to create taxonomies and associate application content with the facets within the taxonomies. Oracle Knowledge contains various classifiers to access both structured and unstructured data sources, as described in [Personalized Navigation Application Classification Components on page 155](#).

## Taxonomy Builders

Taxonomy builders are procedures that you configure within classifiers to create new facets during the classification process. Taxonomy builders enable you to automate the facet creation process, as well as automatically update the facets within data repositories that are subject to frequent change. See [Personalized Navigation Application Taxonomy Builders on page 169](#) for more information on configuring taxonomy builders within classifiers.

## Formatters

Formatters are procedures within taxonomy builders that perform string substitution on facet names so that you can control their presentation within the User Interface. See [Personalized Navigation Application Formatters on page 176](#) for information on configuring formatters within classifiers.

## Comparators

Comparators are modules within taxonomy builders that compare two values and determine whether the first is less than, equal, or greater than the second. See [Personalized Navigation Application Comparators on page 175](#) for more information on configuring comparators within classifiers.

## Implementing Personalized Navigation

You implement Personalized Navigation using the Personalized Navigation application within the Oracle Knowledge Language Workbench to define content categories, and by configuring the User Interface presentation elements.

To implement Personalized Navigation:

- Activate the Personalized Navigation User Interface template, as described in [Activating the Personalized Navigation User Interface Layout on page 104](#).
- Access the Personalized Navigation application, as described in [Accessing the Personalized Navigation Application on page 104](#).
- Select or define a navigation application, as described in [Defining a Navigation Application on page 105](#).
- Define taxonomies and facets within the navigation application, as described in [Defining Taxonomies on page 107](#).
- Create any necessary Dictionary rules to reference taxonomies, as described in [Referring to Facets from Dictionary Rules on page 128](#).
- Classify the application data, as described in [Performing Data Classification on page 127](#).

## Activating the Personalized Navigation User Interface Layout

To implement the Personalized Navigation User Interface elements, you activate the Personalized Navigation User Interface layout, `ui_search_and_browse.xml`, located in:

```
webapps/xml/search/main.xml
```

The `main.xml` file contains an include statement for the Personalized Navigation User Interface layout.

The following is an example of the include statements within the `main.xml` file, showing the Personalized Navigation layout enabled:


```
<!-- Options for Search UI Main Screens -->
<!--xsl:import href="ui_search_basic.xml"/-->
<xsl:import href="ui_search_and_browse.xml"/>
<!--xsl:import href="ui_search_vrep.xml"/-->
```

**Important!** You can enable only one of the include statements for your application.

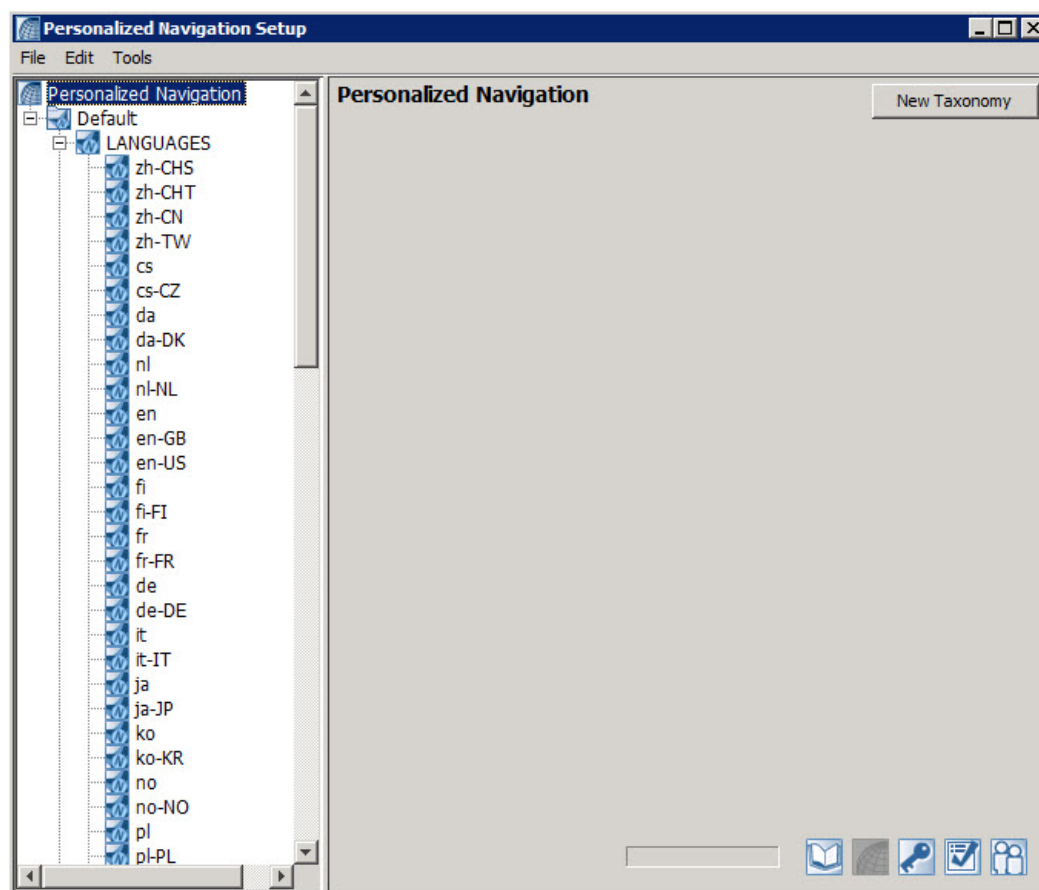
## Accessing the Personalized Navigation Application

You access the Personalized Navigation application:

- From the Oracle Knowledge Language Workbench Launcher.

- Enter a valid User name and password.
- Select **Personalized Navigation Setup** from the Application drop-down menu.
- From other Oracle Knowledge Language Workbench applications, such as the User Manager.
  - Click on the Personalized Navigation application icon , displayed in the lower right corner of the main application window.

The Oracle Knowledge Language Workbench displays the Personalized Navigation application screen:



The Personalized Navigation application displays a tree containing the Default navigation application and any other currently defined applications.

You can select the **Default** folder to define taxonomies within the Default navigation application, or define additional navigation applications.

## Defining a Navigation Application

Navigation applications are collections of content categories, or taxonomies that you define for your Oracle Knowledge application. You can define any number of navigation applications in addition to the Default application.

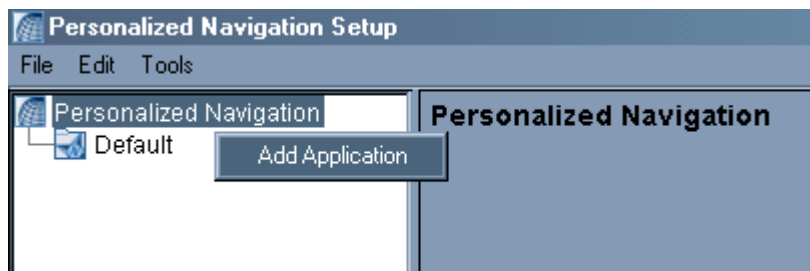
You configure the navigation application for use with your application using the Advanced Configuration Facility, as described in [Configuring the Navigation Application on page 106](#).

**Note:** The Default navigation application is configured by default.

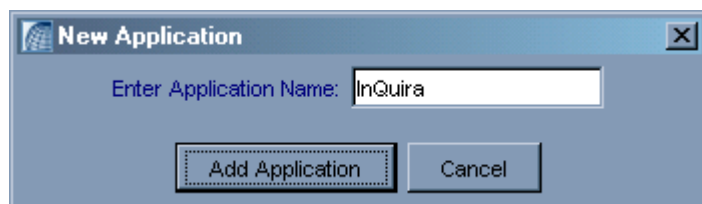
Each navigation application supports any number of top level taxonomies. The top level taxonomies are the highest level categories that the User Interface displays to end users.

To create a navigation application:

- 1 Select **Edit** from the Menu bar, or right-click on the **Personalized Navigation** folder and select **Add Application**:



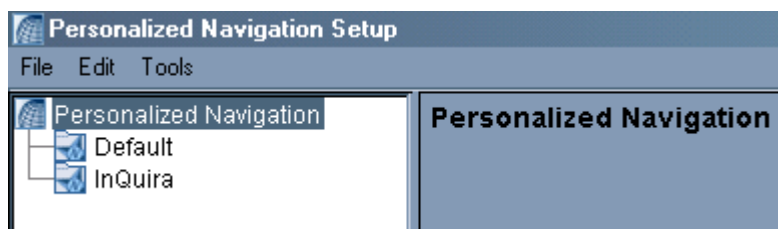
The New Application dialog displays:



- 2 Enter an application name and select **Add Application**.

**Note:** The application name does not appear in the User Interface; the top-level taxonomies are the highest level categories visible in the User Interface.

The new application appears as a sibling of the Default application:



## Configuring the Navigation Application

If you define facets within a navigation application other than the default (Default), you must configure the alternate navigation application for use with the application.

**Note:** The Personalized Navigation application displays alternate navigation applications as siblings of the Default application.

You configure a navigation application on the Service Configuration page of the Advanced Configuration Facility.

To configure a navigation application:

- 1 Select **Service Configurations** from the Advanced Configuration Facility menu.

The Service Configurations page displays the current instance configurations:

Editing: Service Configurations

Service : [Default Configuration](#)

Configurations [Distributed Runtime Query Coordinator Configuration](#)

[QueryWorker Configuration](#)

[Add New Item](#)

OK Cancel

- 2 Select the desired instance.

The Service Configurations page displays parameters for the Navigation Service.

Navigation Service

Collection :

Directory :

Default Facet Collection ID :

- 3 Specify the name of the alternate navigation application in the Default Navigation Application ID field, and select **OK** to save your configuration changes.

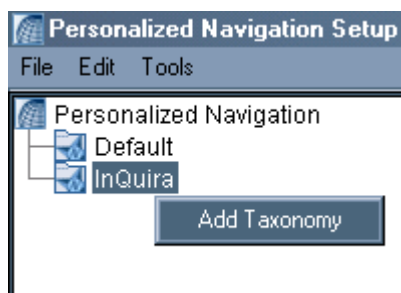
## Defining Taxonomies

You define taxonomies (content categories) for a navigation application using the Personalized Navigation application setup process.

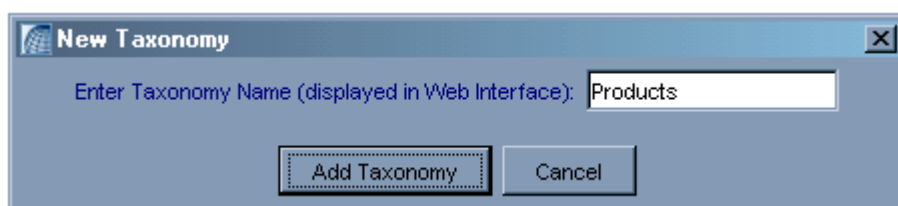
**Note:** Personalized Navigation includes advanced functions for creating and defining content categories that you access using the Show Advanced Setup option in the Tools menu, as described in [Using Advanced Facet Definition Settings](#).

To create a top level taxonomy within a navigation application:

- 1 Select the application folder, then right-click and select **Add Taxonomy** or select the **New Taxonomy** button in the upper right portion of the display:

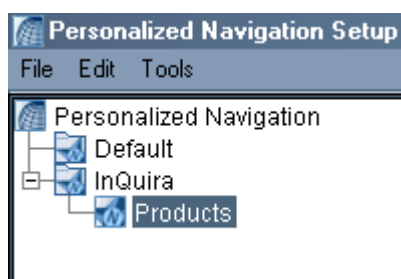


The **New Taxonomy** dialog displays.



- 2 Enter the name for the new taxonomy as you want it to display in the User Interface and select **Add Taxonomy**.

The new taxonomy appears as a node under the selected application:



You can define additional taxonomies as siblings and children of the new taxonomy using the same process.

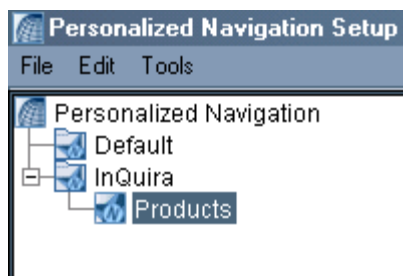
## Defining Taxonomy Display Properties

You can define taxonomies to be displayed in the User Interface, or to be used only internally by the application's request and response processing.

**Note:** Personalized Navigation includes advanced functions for creating and defining content categories that you access using the Show Advanced Setup option in the Tools menu, as described in [Using Advanced Facet Definition Settings](#). The following description applies to the default, and not to the Advanced Setup.

To define the display properties of a taxonomy:

- 1 Select the taxonomy folder for the category:



The Personalized Navigation application prompts you to specify whether this taxonomy:

- Displays in the User Interface and is available for end users to select.
- or -
- Does not display in the User Interface and is available only for reference by Dictionary Rules

1) Do you want this taxonomy displayed in the Web Interface for User Navigation?

☐ Yes -- taxonomy will be displayed for user navigation

☐ No -- taxonomy will only be used by language rules

**Note:** The Advanced setup displays a different screen.

- 2 Select one of the following display options:

<b>Yes</b>	Specifies that this taxonomy displays in the User Interface. The User Interface displays the Web Interface Settings dialog.
<b>No</b>	Specifies that this taxonomy is used only for reference by Dictionary Rules.

The Personalized Navigation application displays the Web Interface Settings dialog for taxonomies that are specified to display in the User Interface:

1) Do you want this taxonomy displayed in the Web Interface for User Navigation?

☒ Yes -- taxonomy will be displayed for user navigation

**Web Interface Settings**

Navigation Area

☐ Show taxonomy before a question is asked.

Maximum number of subfacets displayed before 'more...' link:

Number of facet tree levels to initially display:

Answers Area

List  after each answer.

### 3 Specify the following Navigation Area display options:

#### Show taxonomy before a question is asked.

Select this checkbox to specify that the category displays in the Personalized Navigation column of the User Interface.

For example the categories Price, Year of Publication, and Collections are set to display in the initial question page before a question has been asked:



**Find Answers**

» [Start over](#)

**Price:**

- ▶ [Under \\$0](#)
- ▶ [\\$0 - \\$10](#)
- ▶ [\\$10 - \\$20](#)
- ▶ [\\$20 - \\$30](#)
- ▶ [\\$30 - \\$40](#)
- ▶ [\\$40 - \\$50](#)
- ▶ [Over \\$50](#)

**Year of Publication:**

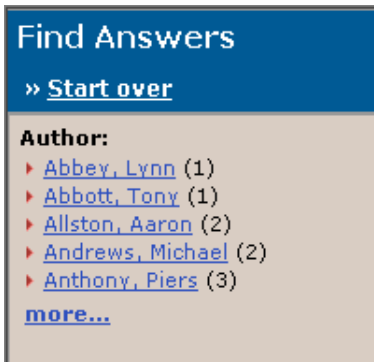
- ▶ [1970-1979](#)
- ▶ [1980-1989](#)
- ▶ [1990-1999](#)
- ▶ [2000-present](#)
- ▶ [Unknown](#)

**Collections:**

- ▶ [Reviews](#)

#### Maximum number of subfacets displayed before 'more...' link:

Specify the maximum number of facets to display in a taxonomy. Additional taxonomies can be displayed by selecting the **more...** link. For example, the following Author taxonomy is limited to five displayed facets:



**Find Answers**

» [Start over](#)

**Author:**

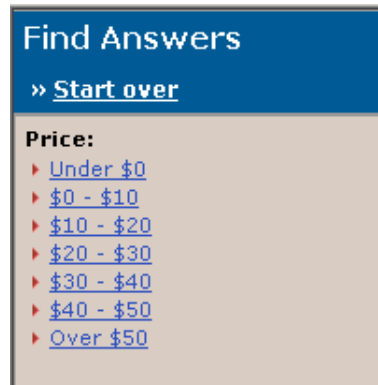
- ▶ [Abbey, Lynn](#) (1)
- ▶ [Abbott, Tony](#) (1)
- ▶ [Allston, Aaron](#) (2)
- ▶ [Andrews, Michael](#) (2)
- ▶ [Anthony, Piers](#) (3)

[more...](#)



**Number of facet tree levels to initially display**

Specify the number of facet levels to display in the initial navigation area. For example, the following Price taxonomy displays two levels, the parent Price, and the first child level of price facets:



#### 4 Specify the following Answer Area display option:

**List <number of facets> after each answer**

Specify the number of facets to include as part of answers belonging to this facet. Select from the following options:

- no facets
- only the first matching facet
- all matching facets

For example, the following answer includes the matching Reviews facet of the Collections taxonomy:



## Specifying Automatic Facet Creation

The Personalized Navigation application prompts you to specify whether it uses the taxonomy builder feature to automatically generate a hierarchy of child facets for this taxonomy:

2) Do you want the system to generate a taxonomy hierarchy for this facet?

- ☐ Yes -- system will generate taxonomy hierarchy below this facet
- ☐ No -- you will manually assign content to this facet

- Select one of the following options:

<b>Yes</b>	Specifies that the Personalized Navigation application generates additional facets within this taxonomy.
<b>No</b>	Specifies that you assign content to this facet.

**Note:** We recommend that you specify to automatically classify content that is stored in a repository, such as a database, that is external to the Content Store.

The Personalized Navigation application prompts you in later steps to specify:

- The source of the content to use.
- Additional generation information, depending on the type of content source you select, as described in [Specifying Generation Options for Structured Content on page 113](#).

## Specifying the Content Source for the Taxonomy

The Personalized Navigation application prompts you to specify whether the content for the taxonomy resides:

- In a database
- In unstructured documents

3) Is the content to be classified in a database?

☐ Yes -- content is in a database table

☐ No -- content is in one or more collections or other documents

- Select one of the following options:

<b>Yes</b>	<p>Specifies that the content for this taxonomy resides in a database.</p> <p><b>Important!</b> The database must be configured as a source of structured data for your application, as described in the <a href="#">Oracle Knowledge Intelligent Search Application Development Guide</a>.</p> <p>The Personalized Navigation application displays the Database Content Settings dialog.</p>
<b>No</b>	Specifies that the content for this taxonomy resides in unstructured content.

For taxonomy content that resides in a database, the Personalized Navigation displays the Database Content Settings dialog:

☒ Yes -- content is in a database table

Database Content Settings

Select the database table to use:

- Select a database table from the list of currently configured schema on the Structured Data page of the Advanced Configuration Facility Instances section

The Personalized Navigation application prompts you to specify generation options.

## Specifying Generation Options for Structured Content

**Note:** This section discusses the Generation Options when a Database table has been chosen as content source. There are different options if the content source for classification is unstructured content.

The Personalized Navigation application presents options to:

- Automatically generate facets from database or document content
- Manually assign database or document content to facets

based on the following options, and the selections you specified in [Specifying Automatic Facet Creation on page 111](#):

If You Chose to....	Go to...
have the Personalized Navigation application automatically generate facets for database content...	<a href="#">Automatically Generating Facets For Database Content on page 113</a>
have the Personalized Navigation application automatically generate facets for document content...	<a href="#">Specifying Generation Options for Unstructured Content on page 117</a>
manually assign content to facets for database content...	<a href="#">Manually Assigning Database Content to a Facet on page 121</a>
manually assign content to facets for document content...	<a href="#">Manually Assigning Document Content to a Facet on page 123</a>

## Automatically Generating Facets For Database Content

If you chose to automatically generate facets for database content, the Personalized Navigation application displays the following generation options:

4) Select one of the following taxonomy generation options:

- ☐ Generate a date taxonomy for a date column
- ☐ Generate a numeric range taxonomy for a numeric column
- ☐ Generate a taxonomy based on the contents of one table column
- ☐ Generate a taxonomy based on the contents of multiple table columns

- Select one of the following options:
  - Generate a date taxonomy for a date column, as described in [Generating a Date Taxonomy for a Selected Column on page 114](#)
  - Generate a numeric range taxonomy for a numeric column, as described in [Generating a Numeric Range \(Price\) Taxonomy on page 115](#)
  - Generate a taxonomy based on the contents of one table column, as described in [Generating a Facet Based on a Single Column on page 116](#)
  - Generate a taxonomy based on the contents of multiple table columns, as described in [Generating a Taxonomy Based on Multiple Columns on page 117](#)

## GENERATING A DATE TAXONOMY FOR A SELECTED COLUMN

This option creates a date taxonomy based on the selected date column and additional parameters.

4) Select one of the following taxonomy generation options:

☒ Generate a date taxonomy for a date column

Settings for generating a date taxonomy

Creates a date taxonomy, using the values of a selected date table column and the date parameters you specify below.

Select the date column to use: start\_time ▼

☐ Generate a 3-level (year/month/day) taxonomy:

☐ Show facet for The current day:

☐ Show facet for the current week:

☐ Show facet for the current month:

☐ Show facet for the current year:

☐ Show facet for all dates:

To generate a date taxonomy for a date column, specify the following parameters:

Field	Description
<b>Select the date column to use</b>	Select a column from the list. The Personalized Navigation application examines the available fields in the specified database and provides a list of columns that contain date data.
<b>Generate a 3 level (year/month/day) taxonomy</b>	Select to automatically generate a three level hierarchy consisting of the facet levels Year, Month, and Day.
<b>Show facet for the current (day, week, month, year):</b>	Select to generate only the specific facets. The Personalized Navigation application displays a field in which you can specify a facet label to override the default.
<b>Show facet for all dates:</b>	Select to generate a single taxonomy of facets for all dates. The Personalized Navigation application displays a field in which you can specify a label to override the default.

See [Classifying Content Based on Database Column Values on page 140](#) for more information on classifying database content.

## GENERATING A NUMERIC RANGE (PRICE) TAXONOMY

This option creates a numeric range taxonomy, such as a price taxonomy, based on a selected numeric column and additional parameters:

4) Select one of the following taxonomy generation options:

☐ Generate a date taxonomy for a date column

☒ Generate a numeric range taxonomy for a numeric column

Settings for generating a numeric range taxonomy

Creates a numeric range taxonomy, using the values of a selected price table column and the range parameters you specify below.

Select the numeric column to use: run\_id

☐ Currency values

☒ Show 'less than minimum' facet: Under

Minimum Value:

Maximum Value:

☒ Show 'greater than maximum' facet: Over

Facet Levels: 2 Facet Level Subdivisions

Level 1	5
Level 2	2

☐ Generate levels beyond level 2 at runtime.

- Specify the following options:

Field	Description
<b>Select the numeric column to use</b>	Select a column from the list. The Personalized Navigation application examines the available fields in the specified database and provides a list of columns that contain numeric data.
<b>Currency values</b>	Select the checkbox to specify that facets sort by currency in ascending order.
<b>Show 'less than minimum' facet:</b>	<p>Select this option to create a facet for values less than the Minimum. The Personalized Navigation application creates a facet labeled:</p> <p style="text-align: center;">Under &lt;minimum_value&gt;</p> <p>for example,</p> <p style="text-align: center;">Under \$5.00</p> <p>where 5 is the value of the Minimum parameter.</p>

<b>Show 'greater than maximum' facet:</b>	<p>Select this option to create a facet for values greater than the Maximum. The Personalized Navigation application creates a facet labeled:</p> <p style="text-align: center;">Over &lt;maximum_value&gt;</p> <p>for example,</p> <p style="text-align: center;">Over \$50.00</p> <p>where 50 is the value of the Maximum parameter.</p>
<b>Facet levels:</b>	<p>Specify the number of levels for the taxonomy, and the number of facets within each level. For example, enter 5, 2 to specify that the top level taxonomy is divided into five facets, and the second level is divided into two facets.</p>
<b>Generate levels beyond level n at runtime</b>	<p>Select the checkbox to create taxonomy levels dynamically at runtime when specifically selected by the user. Use this setting when:</p> <ul style="list-style-type: none"> <li>• The results of conventionally classifying the data are very large.</li> <li>• The performance loss of accessing the data source at runtime is acceptable.</li> </ul>

See [Classifying Content Based on Database Column Values on page 140](#) for more information on classifying database content.

## GENERATING A FACET BASED ON A SINGLE COLUMN

This option creates a taxonomy based on a selected database column. The taxonomy contains a facet for each unique value in the selected column.

4) Select one of the following taxonomy generation options:

- ☐ Generate a date taxonomy for a date column
- ☐ Generate a numeric range taxonomy for a numeric column
- ☒ Generate a taxonomy based on the contents of one table column

Settings for generating a taxonomy from a table column

Creates a single-level taxonomy, using the values of a selected table column. One facet is created for each unique value in the column.

Select the table column to use:

To generate a taxonomy based on a selected column, select a column from the list.

See [Classifying Content Based on Database Column Values on page 140](#) for more information classifying database content.

## GENERATING A TAXONOMY BASED ON MULTIPLE COLUMNS

This option creates a taxonomy based on multiple selected columns. The Personalized Navigation application creates a taxonomy level for each column, in the order specified. You can use the arrows in the Table Columns field to change the order of the columns.

4) Select one of the following taxonomy generation options:

- ☐ Generate a date taxonomy for a date column
- ☐ Generate a numeric range taxonomy for a numeric column
- ☐ Generate a taxonomy based on the contents of one table column
- ☒ Generate a taxonomy based on the contents of multiple table columns

Settings for generating a taxonomy from multiple table columns

Creates a multi-level taxonomy, using the unique values of each selected database table column to create subfacets of the preceding column. The unique values of the first column will appear as subfacets of this facet. Thus specifying 3 columns will result in a 4-level taxonomy including this facet.

Table Columns -- Order-dependent

Select the column(s) to use:

Columns:

profile\_id

question\_set\_id

New Column: question\_set\_id

Add
Delete

To generate a numeric range taxonomy for a numeric column, select the columns to use.

See [Classifying Content Based on Database Column Values on page 140](#) for more information classifying database content.

## Specifying Generation Options for Unstructured Content

If you chose to automatically generate facets for document content, the Personalized Navigation application displays the following generation options:

4) Select one of the following taxonomy generation options:

- ☐ Generate a taxonomy hierarchy for one or more collections.
- ☐ Generate a taxonomy of concepts.
- ☐ Generate a taxonomy using HTML meta tags.

- Select one of the following options:
  - generate a taxonomy hierarchy for one or more collections, as described in [Generating a Taxonomy Based on Directory Structure on page 118](#)
  - generate a taxonomy of concepts, as described in [Generating a Taxonomy from Concepts on page 118](#)
  - generate a taxonomy using HTML meta-tags, as described in [Generating a Taxonomy from Meta-Tags on page 119](#)

## GENERATING A TAXONOMY BASED ON DIRECTORY STRUCTURE

This option creates a taxonomy based on the directory structure of the specified collections.

4) Select one of the following taxonomy generation options:

☒ Generate a taxonomy hierarchy for one or more collections:

Settings for generating a taxonomy from selected collections.

Creates a multi-level taxonomy, using the directory structure of the documents in the selected collections, much as a file browser would do.

Select the collections to use:

Collections:

Collections:

New Collection:

Add Delete Replace

To generate a taxonomy based on one or more specified collections, select the collections to use.

See [Classifying Content by Directory Structure on page 152](#) for more information.

## GENERATING A TAXONOMY FROM CONCEPTS

This option creates a taxonomy based on specified Dictionary concepts. You can specify the following concept criteria:

- Concepts to be automatically discovered by the application:

Enter the Concepts(s) to use:

Auto-detect Concepts

Concepts:

New Concept:

Add Delete Replace

Specify representative concepts for which you want the system to locate and generate taxonomies for similar concepts. You can use the % character as a wildcard. For example, specify:

```
noun.<concept_name>.%
```

to generate a taxonomy based on the hierarchy of noun concepts that are specific to your application.



- Concepts to include as the basis of the taxonomy:

Specify one or more concepts that you want to use as the root of the taxonomy.

- Concepts to exclude from the taxonomy

Specify any concepts that you want to exclude from taxonomies. The Personalized Navigation application excludes the children of excluded concepts unless they are explicitly included as separate entries.

See [Classifying Content Based on Dictionary Concepts on page 151](#) for more information.

## GENERATING A TAXONOMY FROM META-TAGS

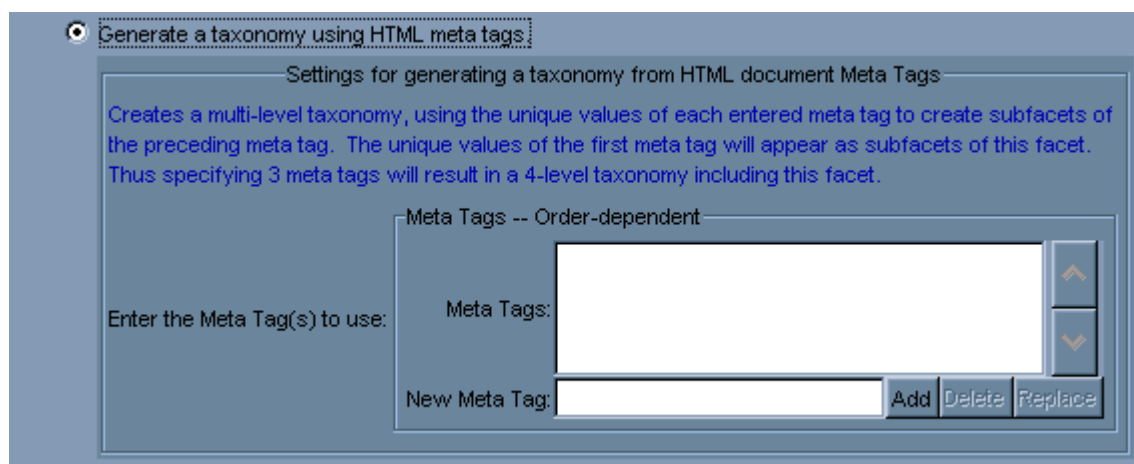
This option creates a taxonomy based on the values of specified HTML meta-tags.

The Personalized Navigation application creates multiple levels based on meta-tags of increasing specificity, for example:

```
<meta name="Genre" content="Fiction"/>
<meta name="SubGenre" content="Speculative"/>
<meta name="SubSubGenre" content="Fantasy"/>
```

Within each taxonomy, it creates a facet for each unique content value within a specified tag.

**Important!** You must ensure that the meta-tags that you specify for classification are added to the Preprocessor's HTML converter configuration file (<InQuira-home>\src\prep\html.config), as described in [Configuring the Preprocessor to Preserve Meta Tags for Classification on page 120](#).



To generate the taxonomy, enter the desired meta-tags.

See [Classifying Content Based on HTML Meta Tags on page 149](#) for information on the advanced Meta Tag classifier settings.

## CONFIGURING THE PREPROCESSOR TO PRESERVE META TAGS FOR CLASSIFICATION

The primary way to configure the preprocessor is through the System Manager/Advanced Config/Preprocessor screens. You can also make manual edits.

For manual edits, you must add the meta-tags that you want to use for Personalized Navigation to the configuration file that controls the behavior of the Oracle Knowledge Preprocessor.

The Preprocessor is a module that prepares application content for use by Oracle Knowledge by translating it from its native format to a standard XML format called Oracle Knowledge Input XML. By default, the Preprocessor removes meta-tags from documents.

To add meta-tags to the Preprocessor configuration:

- Edit the Preprocessor HTML configuration file (html.config):

```
<inquira_home>\src\prep
```

- Add a directive for each meta tag that you want to use in categorization in the form:

```
meta name="<tag_name>" <tab> !meta
```

For example to configure the meta tag:

```
<meta name="Genre" content="Fiction"/>
```

- Specify:

```
meta name="Genre" <tab> !meta
```

The output of this directive is stored in the application's meta-index, in the form:

```
<p>
<meta>[[[GENRE]][[Fiction]]]</meta>
</p>
```

## Manually Assigning Database Content to a Facet

If you chose to manually assign database content to the new facet, the Personalized Navigation application displays the following generation options:

4) Select one of the following facet assignment options:

- ☐ Assign all table entries to the selected facet.
- ☐ Assign selected table entries to the selected facet.

- Select one of the following options:
  - Assign all table entries to the facet, as described in [Assigning All Table Entries to a Facet on page 121](#)
  - Assign selected table entries to the facet, as described in [Assigning Selected Tables to a Selected Facet on page 121](#)

### ASSIGNING ALL TABLE ENTRIES TO A FACET

This option creates a facet (single taxonomy level) containing all table entries for the specified datasource.

4) Select one of the following facet assignment options:

- ☒ Assign all table entries to the selected facet.
- Settings for assigning all table entries to the facet:
- Include all table entries in the selected facet.

See [Classifying Content Based on Database Column Values on page 140](#) for more information on classifying database content.

### ASSIGNING SELECTED TABLES TO A SELECTED FACET

This option creates a taxonomy level containing all entries in the specified data source that match the specified SQL condition.

4) Select one of the following facet assignment options:

- ☐ Assign all table entries to the selected facet.
- ☒ Assign selected table entries to the selected facet.

Settings for assigning selected table entries to the facet:

Include all table entries that match the given SQL 'where' clause.

Enter an SQL 'where' clause to select entries from table:

To assign selected table entries to a selected facet enter a condition to select table entries using any valid SQL to construct a `where` clause, such as:

```
where <field> <comparison> <value>
```

where:

field(s)	Specifies one or more fields from which to select.
comparison	Specifies how to compare the field value with the specified value, for example =, <, >.
value	Specifies the value to compare against to determine selection, for example 5.

**Example:**

where id=5

See [Using an SQL Condition to Classify Content on page 139](#) for more information.

## Manually Assigning Document Content to a Facet

If you chose to manually assign document content to the new facet, the Personalized Navigation application displays the following generation options:

4) Select one of the following facet assignment options:

- ☐ Assign document collections to the selected facet.
- ☐ Assign documents by their type to the selected facet.
- ☐ Assign documents by their language to the selected facet.
- ☐ Assign documents by their content to the selected facet.
- ☐ Assign documents by their URL to the selected facet.

- Select one of the following options:
  - Assign document collections to the facet, as described in [Assigning Documents to a Facet by Collection on page 123](#)
  - Assign documents to the facet by type, as described in [Assigning Documents to a Facet by Type on page 124](#)
  - Assign documents to the facet by language, as described in [Assigning Documents to a Facet by Language on page 125](#)
  - Assign documents to the facet by content, as described in [Assigning Documents to a Taxonomy by Content on page 125](#)
  - Assign documents to the facet by URL, as described in [Assigning Documents to a Facet by URL on page 126](#)

### ASSIGNING DOCUMENTS TO A FACET BY COLLECTION

This option assigns all documents within the specified collections to this facet (single taxonomy level).

4) Select one of the following facet assignment options:

☒ Assign document collections to the selected facet.

Settings for assigning collections of documents to the facet

Include all documents in the selected document collection(s).

Select the collection(s) to include:

Collections:

Collections:

New Collection:

To assign document collections to a selected facet, enter one or more collections. The order in which you list the collections does not affect composition of the taxonomy level.

See [Classifying Content by Collection on page 146](#) for more information.

## ASSIGNING DOCUMENTS TO A FACET BY TYPE

This option creates a taxonomy consisting of facets for each selected document type. You can select one or more of the following document types:

- HTML
- IQXML
- MS-EXCEL
- MS-POWERPOINT
- MS-WORD
- PDF
- RTF
- TEXT
- UNKNOWN
- XML

4) Select one of the following facet assignment options:

☐ Assign document collections to the selected facet.

☒ Assign documents by their type to the selected facet.

Settings for assigning documents to the facet based on their type

Include all documents of the selected type(s).

Select the document type(s) to include:

Document Types

Document Types:

New Document Type: HTML

To assign document collections to a selected facet, specify one or more document types.

See [Classifying Content by Document Type on page 147](#) for more information.

## ASSIGNING DOCUMENTS TO A FACET BY LANGUAGE

This option creates a taxonomy containing facets for each specified language. Each facet contains only documents identified as belonging to the specified language.

4) Select one of the following facet assignment options:

- ☐ Assign document collections to the selected facet.
- ☐ Assign documents by their type to the selected facet.
- ☒ Assign documents by their language to the selected facet.

Settings for assigning documents to the facet based on their language

Include all documents written in the selected language(s).

Select the language(s) to include:

Document Languages

Languages:

New Language: Traditional Chinese Add Delete

To assign documents to the facet by language, add one or more languages.

See [Classifying Content Based on Language on page 148](#) for more information.

## ASSIGNING DOCUMENTS TO A TAXONOMY BY CONTENT

This option assigns content to this taxonomy based on InQuira Match Language (IML) expressions. See [Chapter 4, IML Reference](#) for more information on using IML.

4) Select one of the following facet assignment options:

- ☐ Assign document collections to the selected facet.
- ☐ Assign documents by their type to the selected facet.
- ☐ Assign documents by their language to the selected facet.
- ☒ Assign documents by their content to the selected facet.

Settings for assigning documents to the facet based on their content

Include all documents for which the given search expression matches the portion of the document specified by the 'search scope'. Note: the search expression can be any word or valid IML expression.

Enter text to match (any IML expression is valid):

Select the scope of the search within the document: Document

To assign documents to the facet by content:

- 1 Specify an IML expression to match the desired content.
- 2 Select document scope you want to apply the expression to.

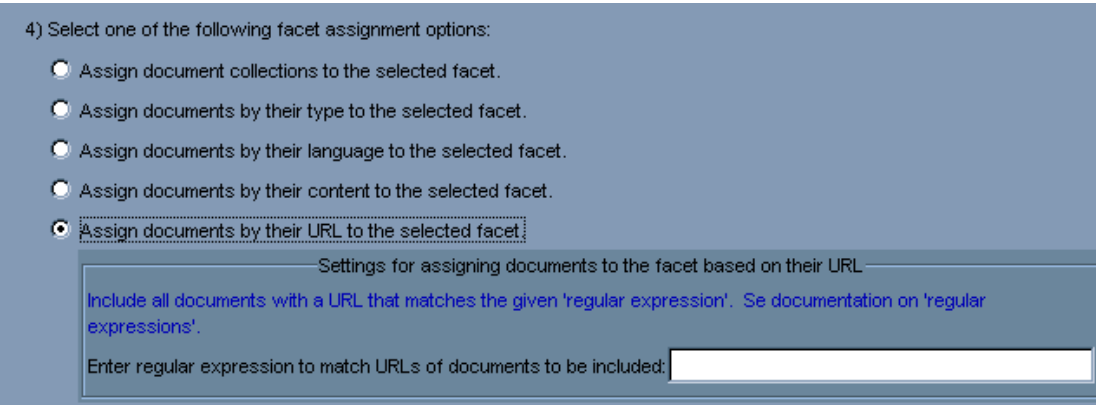
For example, you could specify the IML expression `cat OR dog` and the `Document` scope to match all documents that contain the words `cat` or `dog`.

Similarly, you could specify the `Sentence` scope to match only sentences that contain the words `cat` or `dog`. Or, you could specify `Title` scope to match only documents that contain the words `cat` or `dog` in their title.

See [Using IML Expressions to Classify Content on page 148](#) for information on the advanced IML classifier settings.

## ASSIGNING DOCUMENTS TO A FACET BY URL

This option assigns content to this taxonomy based on regular expression matches to a document URL. See [Specifying Regular Expression Substitutions on page 175](#) for more information on specifying regular expressions to match document URLs.



4) Select one of the following facet assignment options:

- ☐ Assign document collections to the selected facet.
- ☐ Assign documents by their type to the selected facet.
- ☐ Assign documents by their language to the selected facet.
- ☐ Assign documents by their content to the selected facet.
- ☒ Assign documents by their URL to the selected facet.

Settings for assigning documents to the facet based on their URL

Include all documents with a URL that matches the given 'regular expression'. See documentation on 'regular expressions'.

Enter regular expression to match URLs of documents to be included:

To assign documents to a taxonomy by URL, specify a regular expression to match the desired URL.

See [Classifying Content by URL on page 153](#) for more information.



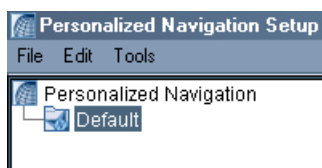
## Performing Data Classification

Once you have configured data classification, you operate the classification process to build the taxonomies for your application using the Personalized Navigation application.

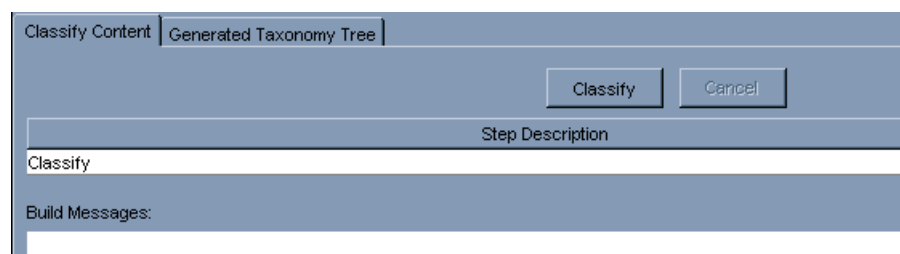
The classification process associates the configured taxonomies with the actual application content, and stores the classification information in the application index.

To perform the classification operation:

- 1 Select the navigation application configured for your application



- 2 Select Classify on the Classify tab



When the Classify operation is complete, the system automatically displays the Generated Taxonomy tab, showing the results as a tree.

You can also operate the classification process by:

- Configuring a scheduled task, as described in the [Oracle Knowledge Intelligent Search Administration Guide](#)
- Executing the following command on the command line:

```
% scheduler run Navigation -f <navigation application list>
    where <navigation application list> specifies one or more navigation applications to
    classify. For example, you could execute the command

    scheduler(.sh/.bat) run Navigation -f Default.
```

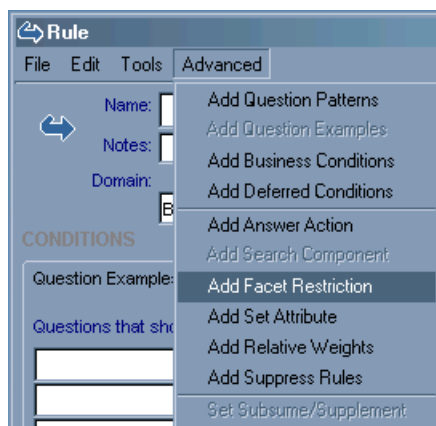
**Important!** You must execute the Key Map command prior to classification if you have taxonomies that reference structured data sources, the referenced data sources are not configured for hybrid search, and the data source has changed since the last classification process.

```
scheduler run KeyMap -d <datasource list>
```

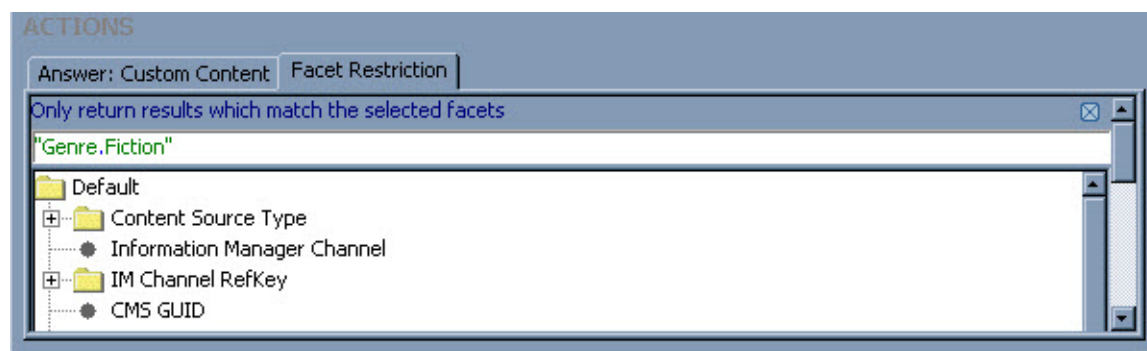
The Key Map command is also available as a scheduled task in the Advanced Configuration Facility.

## Referring to Facets from Dictionary Rules

You can configure a rule to select one or more content categories during request processing using the Advanced Add Facet Restriction function:



The Dictionary Manager displays the Facet Restriction tab within the Rule window:



The Facet Restriction tab displays the currently configured navigation application.

## Referring to Facets by Name

You refer to facets within rules by their full facet names. A full facet name is a string constructed from the names of the facets that compose the path to the desired facet from the root of the taxonomy. Each facet name within the string is separated by a period. For example, the string:

```
Genre.Fiction.Mystery
```

refers to the facet `Mystery` within `Fiction`, within the root taxonomy, `Genre`.

**Note:** You must use an escape character (`\`) to denote a period within a category name.

You can use the wildcard character (`*`) to refer to content categories. For example, the reference:

```
Genre.*.Fantasy
```

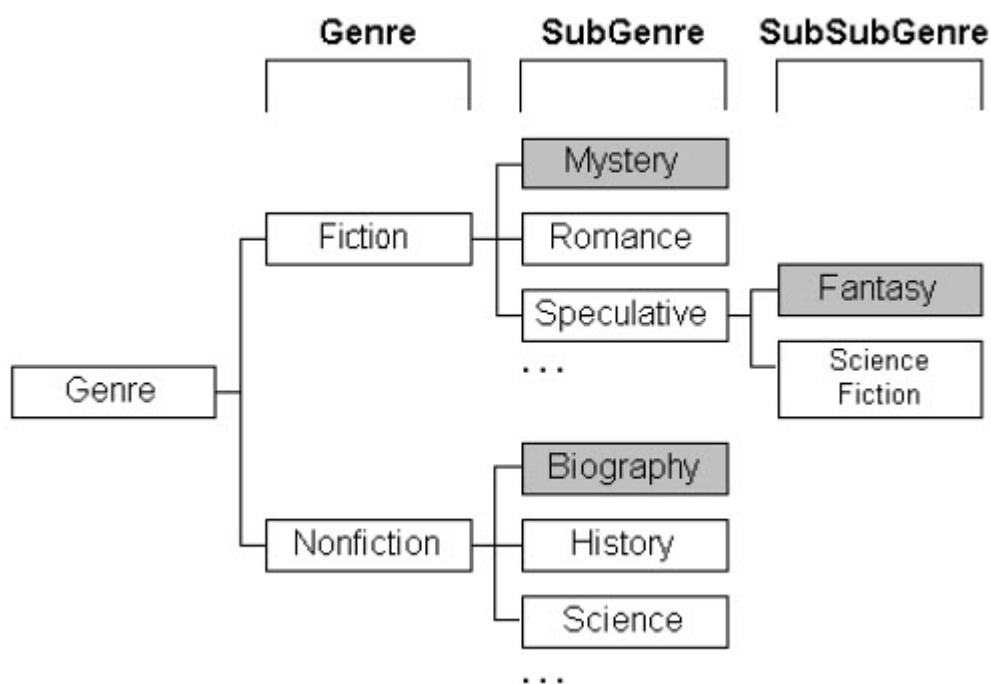
matches the first category in the taxonomy `Genre` that has the name `Fantasy` using a standard breadth first search (the first, shallowest possible match). If there is no match, the category is removed from the expression.

## Selecting Multiple Facets from the Tree Display

You can select multiple facets in the tree display using the Shift and Control (Ctrl) keys to specify:

- The intersection of the taxonomies and
- A union of the facets within the taxonomies

For example, selecting the facets Mystery, Fantasy, and Biography:



produces the expression:

```
("Genre.Fiction.Mystery" OR "Genre.Fiction.Speculative.Fantasy") OVERLAP
"Genre.Non Fiction"
```

## Selecting Multiple Facets using IML

You can use IML to select multiple facets within a single action. For example, you can specify to select two facets, or only results belonging to multiple facets.

To select two facets, `facet_a` and `facet_b`, specify:

```
"facet_a" OR "facet_b"
```

To select results that belong to both `facet_a` and `facet_b`, specify:

```
"facet_a" OVERLAP "facet_b"
```

**Note:** The expression `"facet_a" IS "facet_b"` returns only exact matches for document scope. For example, it correctly returns all documents that were classified as `facet_a` and `category_b`; however, it does not return a document classified as `category_a` and containing a section classified as `category_b`, as does the OVERLAP operator.

## Using VIL in Rules that Select Content Categories

You can use VIL within expressions that refer to specific values within numeric or range-based facets, such as individual prices within a price range facet.

For example, a rule that matches prices and creates a category restriction using VIL similar to:

```
Price.*.{price}.
```

instantiates based on the individual price in the question, but never matches category names expressed as ranges, such as \$5-\$9.99.

To match a specific value with the appropriate range category, specify:

```
Price.<>*. {price}
```

**Note:** This specification requires that the content category have defined minimum and maximum values and a comparator. Some classifiers automatically specify these parameters.

## Restricting the Display for Facets Above a Selected Facet

You can specify additional facet restriction within rules to override the default User Interface behavior, which displays the higher level facets of selected facets, so that users can expand the scope of the results.

To restrict users from accessing higher level facets, use the RESTRICTION parameter:

```
"RESTRICTION(facet_name) "
```

**Important!** RESTRICTION is special syntax within the Personalized Navigation functionality, and is not an IML operator.

## Preventing a Facet from Displaying

You can specify a rule to remove a facet from the display using the ISNT operator in the form:

```
"<facet_parent>" ISNT "<facet_parent.facet_child>"
```

For example, to prevent the Non Fiction facet from displaying, specify:

```
"Genre" ISNT "Genre.Non Fiction"
```

## Specifying Approximate Facet Matching

You can specify to match facets based on an approximate value in the question using the ~\* wildcard. The wildcard specifies to first search for a facet that completely contains the value referred to in the question. If no containing facet is matched, the wildcard specifies to match the parent facet.

For example, the question:

```
Show me books published around 1998
```

might match a date facet selection rule:

```
Year.<>*.1998
```

to return the facet:

```
1998
```

You can use the wildcard to specify:

```
Year.~*.1998
```

to return the facet:

```
1990 - 1999
```

**Note:** This function requires that the facets have minimum and maximum values, and a comparator specified, as described in [The Range Taxonomy Builder on page 173](#) and [The Personalized Navigation Application on page 101](#).

## Selecting a Range of Content Categories

You can specify to select a range of facets within a single rule using the \* IML operator in the format:

```
"Price.<>*. $0" * "Price.<>*. $22"
```

For example, the question:

```
What books cost between $0 and $22
```

matches the following restriction:

```
"Price.<>*. $0" * "Price.<>*. $22"
```

to select the facets:

```
$0-$9.99
$10-$19.99
$20-$29.99
```

**Note:** This function requires that the facets have minimum and maximum values, and a comparator specified, as described in [The Range Taxonomy Builder on page 173](#) and [The Personalized Navigation Application on page 101](#).

## Responding to Requests for Non-Existent Content Categories

You can specify to select the best possible facet for questions that request non-existent data using the greater-than (>\*) and less-than (<\*) wildcards.

For example, the request:

```
Show me books published after 1945
```

does not match any facets for data starting at 1960. You can use the wildcard:

```
<*
```

to match the closest facet greater than the requested value.

## Facet expansion in addition to restriction

Personalized Navigation supports navigation to back out of a selected category to any higher level category. More importantly, users can expand the scope of a natural language question. For example, given the question:

```
What books were published in 1992?
```

You can select a facet to expand your scope to 1990-1999.

You can also expand the scope of unstructured searches. For example, consider a question about car models:

Special deals for Civics?

followed by selecting the Sedans facet.

The application expands the scope of the question to mean:

Special deals for (all sedans)?

even though the original question text still references Civics.

## Understanding Facet Counts

The facet counts are how many documents match the facet, determined by the classifiers in the taxonomy.xml set up in the Personalized Navigation Tool in Workbench, for the candidate set of the question asked. Facet counts include all structured and unstructured answers for all purposes, by default, but does not include managed answers.

The facets are intended to help the user see what other results are available so they can better filter their results. They are not intended as a navigation tool, to see all results matching a facet for any question.

The number in the parentheses next to a facet shows the minimum number of results that could be a match when the facet is selected. This number is based on the candidate set for a question asked, rather than the number of search results shown to the user, referred to here as the result set. The candidate set is the set of documents that the system determines are a match for the question, which is most often larger than the result set displayed to the user. In fact, the candidate set is a superset of the result set.

**Note:** The user may not see all results depending upon the maximum answers displayed. Also, there may be more results depending upon the search algorithm that builds matches.

The candidate set of documents is based on the search component rules in the system. Using the search component rules, the system populates the candidate set with at least as many answers as the result set, assuming there are that many matching answers. The system starts at the top importance level and proceeds down the other importance levels until it fills up the result set. Once the result set size is met, the engine does not proceed to the next search component importance level. So, if the result set size is met with just Required importance level matches, the candidate set is populated with all the Required matches and it does not pull the Very Important, Important, etc. importance level matches. This assumes that the facet setting guarantee-max-results is set to false, the default setting.

If a document is not in this candidate set, it is impossible for it to be returned by the search algorithm as an answer; all possible answers must come from the candidate set. It is this candidate set that is used to create the facet counts. Changing the minimum threshold does not change the candidate set, and therefore not change the facet counts.

## Using Advanced Facet Definition Settings

The Personalized Navigation application provides additional advanced settings for defining and creating taxonomies. You can display the advanced settings for new or existing taxonomies by selecting the Show Advanced Setup option of the Personalized Navigation application tools menu.

**Note:** Selecting the Show Advanced Setup option at any point in the simple Personalized Navigation dialog process displays the corresponding advanced setup. You can toggle between the dialog and the advanced setup at any time. For example, selecting the Show Advanced Setup option immediately after creating a new taxonomy displays the Basic Options in a forms format instead of the simple dialog format, and makes available the Advanced settings.

In the following graphic, Basic Options are noted with **B**.

**Taxonomy Facet: Intelligent Search Guides** New Taxonomy

Facet Settings **Classifiers**

☒ Enabled

Facet -- Settings

**B** ☐ Hidden  
☐ Show Empty Children  
☒ Document Level Count  
☒ Include Parent's Rules  
**B** ☒ Classify Results (All Possible)  
☐ Include generic higher level facets

**B** ☐ Classify Structured Results  
☒ Show Siblings  
☒ Stop at One  
☐ Ignore Unknown Facets  
☒ Classify Results (Only Subselected)  
☐ Display generic higher level facets

**B** ☒ Classify Unstructured Results  
☒ Show Sibling Counts  
☐ Expand Rule Selected Facets  
☒ Allow Facet Expansion  
☐ Automatic  
☐ Classify Results (All generic facets)

**B** Max Children:  **B** Display Depth:   
 Maximum Count Per Classification:  Minimal parent depth for unknown generic facets (1 = taxonomy, etc.):

**B** Label:   
 Arbitrary Data:   
 Minimum Value:   
 Maximum Value:

LEAF\_IN\_COLLECTION: Unknown optionType: gnu.trove.TIntHashSet

Comparators:

Not for Counting

Values:

New Value:  Add Delete Replace

Language Based Label

Language	Label
British English	<input type="text"/>

Add Delete

Rules

Domain:

Rules:

- Define PSEUDO\_TOKEN\_MOD
- ImportantConcept SC
- RequiredConcept SC
- VeryImportantConcept SC

Rule	Domain

Delete



## Specifying Advanced Facet Display Settings

The Advanced Facet Display Settings tab provides advanced options to control whether and how each facet that you create displays in the User Interface.

Basic Display Option	Description
<b>Enabled</b>	Use this setting to specify whether this facet is used to classify data. Facets are enabled by default. Deselect the checkbox to exclude this facet from the classification process.
<b>For Initial Browse</b>	Use this setting to specify whether this facet displays on the initial page before users ask a question or select a facet. Facets are included in the initial browse by default. Deselect the checkbox to specify that the facet does not display on the initial page.
<b>Hidden</b>	Use this setting to specify whether this facet and its children are excluded from the User Interface display and available only to Dictionary rules. Select the checkbox to exclude the facet and its children from the User Interface.
<b>Classify Structured Results</b>	Use this setting to specify whether structured results include the most specific facet matched by that excerpt in this taxonomy. Select the checkbox to include the most specific facet matched for this excerpt in the answer display.
<b>Classify Unstructured Results</b>	Use this setting to specify whether unstructured results include the most specific facet matched by that excerpt in this taxonomy. Select the checkbox to include the most specific facet matched for this excerpt in the answer display.
<b>Classify Results (All Possible)</b>	Use this setting to specify whether results include all the most specific facets in this taxonomy. Select the checkbox to include all specific facets matched for this excerpt in the answer display.
<b>Max Children</b>	<p>Use this setting to specify the maximum number of children of this facet to display before the end user must explicitly request more by selecting a <b>more...</b> link. The default is 20.</p> <p>Specify an integer as the maximum number of children to display.</p>
<b>Display Depth</b>	<p>Use this setting to specify the number of levels in the taxonomy that should be displayed to the end user. The default is 2.</p> <p>Specify an integer as the number of taxonomy levels to display.</p>
<b>Label</b>	<p>Use this setting to specify the label to display in the User Interface for this facet.</p> <p><b>Note:</b> You can specify language-dependent labels using the Language-based Label advanced setting.</p>

The Advanced Options provide greater control for facet display.

Advanced Display Option	Description
Show Empty Children	Use this setting to specify whether to display the children of this facet when no results are available. The default is to exclude empty children from the display. Select the checkbox to display empty child facets.
Show Siblings	Use this setting to specify whether to display siblings of this facet when it is selected. The default is to include the sibling facets in the display. Deselect the checkbox to exclude sibling facets from the display.
Show Sibling Counts	Use this setting to specify whether siblings of this facet display with item counts. The default is to include these counts.  <b>Note:</b> These counts only display when browse mode (no user question) is in effect. Clear the checkbox to exclude sibling counts from the display.
Document Level Count	Use this setting to count a document only once if it contains more than one answer in the document. The default is to count each answer. Deselect the checkbox to count multiple answers in a document as a single value.
Stop at One	Use this setting to specify whether users can drill down by selecting this facet even if there is only one result. The default is to prevent drilling into a single result. Deselect the checkbox to enable users to drill into this facet when it displays only one result.
Expand Rule Selected Facets	Use this setting to specify whether facets selected by the Dictionary rules are expanded as if they were selected explicitly by the user. The default is to leave the rule-selected facets un-expanded. Select the checkbox to expand rule-selected facets.
Include Parent's Rules	Use this setting to specify whether rules associated with the parent facet are evaluated in addition to this facet's rules. The default is to evaluate the parent facet's rules. Deselect the checkbox to restrict evaluation to the current facet.
Ignore Unknown Facets	Select to have unknown facets ignored by language rules while processing searches. This prevents broken logic from returning zero search results.  <b>Note:</b> This value is required only for facets that are referenced by Dictionary rules that perform approximate matching.
Allow Facet Expansion	Select to have the navigation facet expanded in the UI. When a user performs a search and a matching rule contains a facet restriction, selecting this option allows for the taxonomy to be expanded to the level of the facet restriction in the UI navigation.  If in taxonomy T, there is a facet Y and a facet X such that Y is a parent of X (Y → X) and there is a facet restriction rule that restricts results to facet X, clicking on Facet Y in navigation removes the restriction to X.  This feature works for cases where the rule based restriction is on a sub-facet and a higher facet is selected in the UI. For example, if Product A facet has defined subfacets A1 and A2 with facet restriction rules on A1. If <b>Allow Facet Expansion</b> on Product A is turned on, then clicking on Product A will remove the facet restriction rule on A1 and both the sub facets A1 and A2 will be displayed.

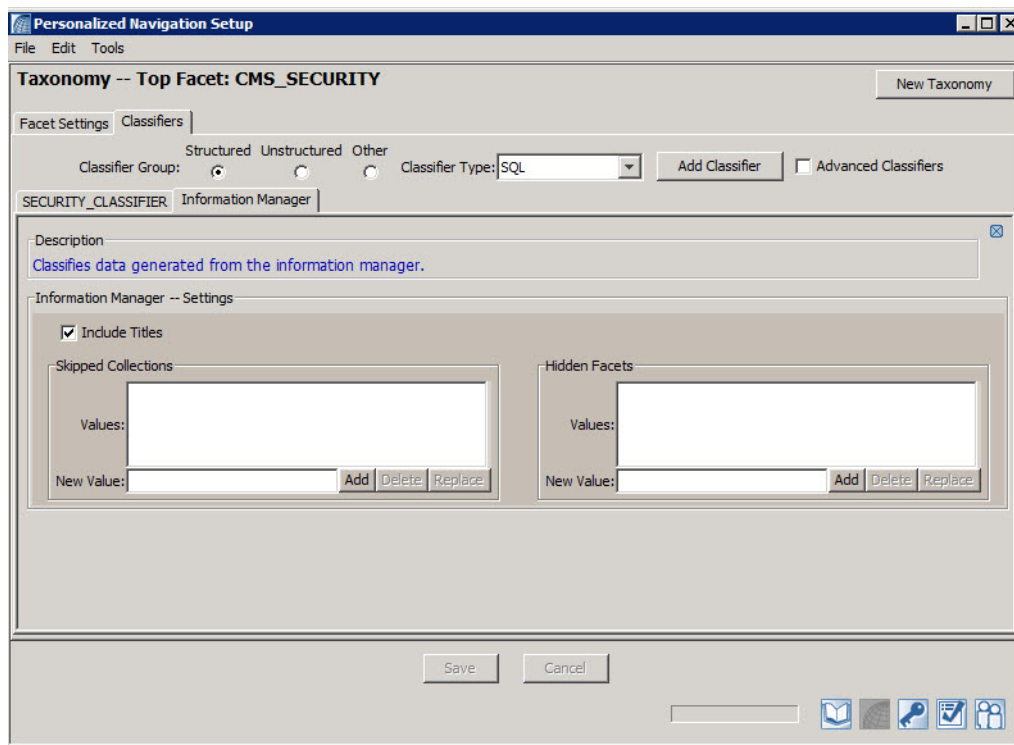
Advanced Display Option	Description (Continued)
<b>Classify Results (Only Subselected)</b>	Select this to perform facet-level excerpting for the lowest level facet.
<b>Automatic</b>	When selected, this indicates that Information Manager created the facet and dynamically populates it.
<b>Include generic higher level facets</b>	Select this to add a <code>_GENERIC</code> facet under each of the parent nodes. The generic facet includes all the documents of the children. Results from generic facets are included in search results.
<b>Display generic higher level facets</b>	Related to the preceding option, in addition to being included in search results, this allows the generic facets to display in the UI as well.  For example, if you restrict to something like <code>Software.Oracle Knowledge.8x</code> , by default, only results that are classified that way display. However, there may be a support document that applies to all versions of Oracle Knowledge, but if you restrict to <code>Software.Oracle Knowledge.8x</code> , then it does not display, because it is not as specific. By including the generic facets, results that are applicable to all Oracle Knowledge, and all Software items display. But results that apply to <code>Software.Windows</code> do not display, because these are not on the same set of branches that you have followed.
<b>Classify Results (all generic facets)</b>	Select to display all generic facets in the UI.
<b>Maximum Count Per Classification</b>	Use this setting to specify the maximum number of results to be counted for a particular piece of classified unstructured data. The default is the maximum number of results allowed to be returned for a document.
<b>Minimal parent depth for unknown generic facets</b>	Enter the minimum parent depth for creating generic facets if <b>Include generic higher level facets</b> is selected. The higher the number, the deeper into the taxonomy generic facets are created. Default is 1.
<b>Arbitrary Data</b>	Use this to send any static XML data for use on the UI.
<b>Minimum Value</b>	Use this setting to specify the smallest value for this facet.  <b>Note:</b> This setting is required only for facets that are referenced by Dictionary rules that perform approximate matching. You can specify any type of value that can be compared, such as numbers, currencies, names, etc.  <b>Note:</b> Some taxonomy builders set these values automatically. If they are not set, then Personalized Navigation uses the facet name for both values.
<b>Maximum Value</b>	Use this setting to specify the maximum value for this facet.  <b>Note:</b> This setting is required only for facets that are referenced by Dictionary rules that perform approximate matching. You can specify any type of value that can be compared, such as numbers, currencies, names, etc.  <b>Note:</b> Some taxonomy builders set these values automatically. If they are not set, then Personalized Navigation uses the facet name for both values.

Advanced Display Option	Description (Continued)
<b>Comparators</b>	<p>Use this setting to specify the comparison method to determine the basis on which the facet items are ordered in the display. Personalized Navigation uses the comparator to compare the requested value with the minimum and maximum settings.</p> <p><b>Note:</b> This value is required only for facets that are referenced by Dictionary rules that perform approximate matching.</p> <p>Select one of the following comparators to specify the basis on which the facets are ordered in the display:</p> <ul style="list-style-type: none"> <li>• Alphabetical</li> <li>• Number</li> <li>• Currency</li> <li>• Dimension</li> <li>• Last Name</li> <li>• Month</li> </ul> <p>See <a href="#">Personalized Navigation Application Comparators on page 175</a> for more information on Comparators.</p>
<b>Not for Counting</b>	<p>Use this setting to specify data sources that should not contribute to facet item counts, which are displayed within parentheses after the facet name. Enter each data source name in the New Value field, and select Add.</p>
<b>Language Based Label</b>	<p>Use this setting to specify labels to display for this facet in languages other than the default application language. Select a supported language from the list, enter the corresponding label, and select Add.</p>
<b>Rules</b>	<p>Use this setting to specify Dictionary rules whose actions are evaluated if this facet is selected. Select the Domain to filter the list of rules, then drag and drop the desired rule into the Rule / Domain field.</p>

## Using Advanced Data Classification

The Classifiers tab in the Personalized Navigation application advanced settings displays options for specifying data classification. You specify classification for a selected facet by configuring one or more classifiers to:

- Identify the properties of the data that you want to include in the current facet
- Identify the properties of the data that you want to include in generated facets that are the children and sub-children of the current facet

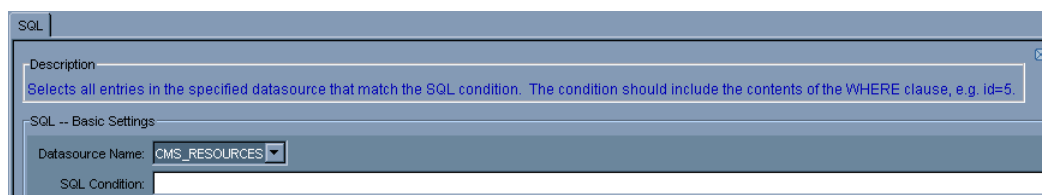


The Personalized Navigation application contains classifiers to access data from structured and unstructured application content.

## Using an SQL Condition to Classify Content

You can use the SQL classifier to classify content based on an SQL condition. This classifier selects all entries in the specified data source that match the specified SQL condition.

**Note:** This classifier does not generate child facets. There is no taxonomy builder associated with this classifier.



You specify the SQL condition using any valid SQL to construct a `where` clause, such as:

```
where <field> <comparison> <value>
```

where:

<code>field(s)</code>	Specifies one or more fields from which to select.
<code>comparison</code>	Specifies how to compare the field value with the specified value, for example: <code>=</code> , <code>&lt;</code> , <code>&gt;</code>

value	Specifies the value to compare against to determine selection, for example 5.
-------	---

### Example:

where id=5

This classifier formats the column values as specified in the default database column formatter, so that they match the corresponding display in the results table. You can achieve more control over formatting structured facets using the Structured Column classifier

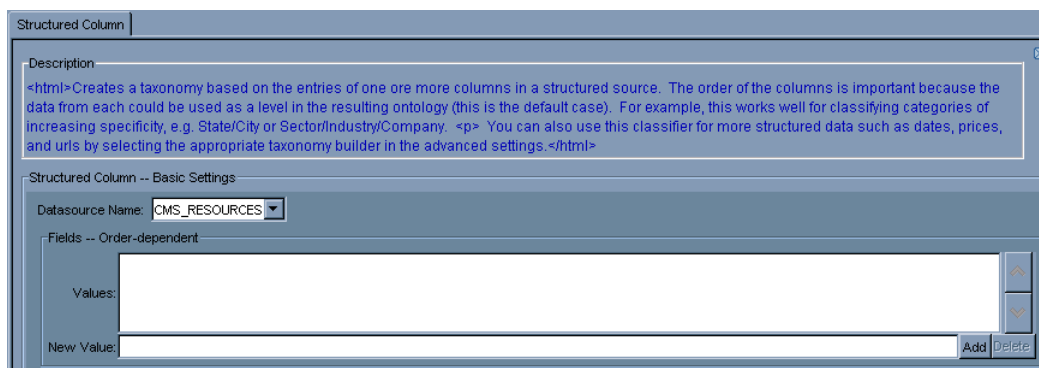
See [The SQL Classifier on page 155](#) for detailed descriptions of the SQL classifier settings.

## Classifying Content Based on Database Column Values

The Structured Column classifier creates a taxonomy based on the entries of one or more columns in a structured datasource.

You can use this classifier to create multiple taxonomy levels of increasing specificity, for example, Country -> State -> City, or Sector -> Industry -> Company.

You can also use this classifier with the appropriate taxonomy builder to create taxonomies for dates, prices, and URLs.



You configure a column classifier by specifying the data source name and the column names that you want to capture. The Structured Column classifier creates a facet for each unique value within the specified column, for example, each unique model name in the `model_name` column. It also creates a separate level for each column that you specify.

**Note:** You may receive SQL errors if one or more of the columns you specified does not exist in the specified schema or table.

You can use additional Structured Column classifier features to:

- Configure multiple taxonomy levels from structured columns
- Configure the appearance of the child facet names
- Specify an optional View for the database

See [The Structured Column Classifier on page 156](#) for detailed descriptions of the Structured Column classifier settings.

## Configuring Multiple Taxonomy levels from Structured Columns

You can configure multiple taxonomy levels by specifying multiple column names. The column classifier uses each column in the list to create child facets of the preceding column, so that specifying three columns results in three taxonomy levels.

**Note:** You may receive SQL errors if one or more of the columns you specified does not exist in the specified schema or table.

## Configuring the Appearance of Structured Column Child Facet Names

You can configure the appearance of the child facet names by specifying regular expressions to perform operations on the column values, such as capitalizing names and removing unwanted characters. See [Specifying Regular Expression Substitutions on page 175](#) for more information on using regular expression replacement.

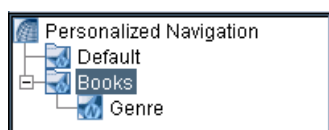
## Example: Creating a Taxonomy from Database Categories

In this example, we create a taxonomy from data stored in a database using the structured column classifier.

Consider a database of information about books that contains following categories, or columns:

```
Genre
SubGenre
SubSubGenre
```

Assume that we have defined a navigation application Books, and added a Genre taxonomy:



To configure the structured column classifier for the Genre taxonomy:

- 1 Select **Genre**, and select the **Classifiers** tab  
The Personalized Navigation application interface displays the Classifiers screen.
- 2 Select the **Structured Column** classifier type from the Structured Classifier group, and select **Add Classifier**  
The Structured Column classifier tab displays.
- 3 Configure the structured column classifier as follows:

Parameter	Configuration
<b>Datasource Name</b>	Select the desired data source from the list of data sources configured for your application.

<b>Fields</b>	<p>Use the New Value field to enter the field names in the order in which they occur in the taxonomy:</p> <p>Genre SubGenre SubSubGenre</p>
<b>Taxonomy Builder</b>	<p>The Column classifier uses the default String taxonomy builder to create child facets from the values contained in the specified columns.</p>

Structured Column

Description

<html>Creates a taxonomy based on the entries of one or more columns in a structured so could be used as a level in the resulting ontology (this is the default case). For example, this State/City or Sector/Industry/Company. <p> You can also use this classifier for more structu taxonomy builder in the advanced settings.</html>

Structured Column -- Basic Settings

Datasource Name: CMS\_RESOURCES

Fields -- Order-dependent

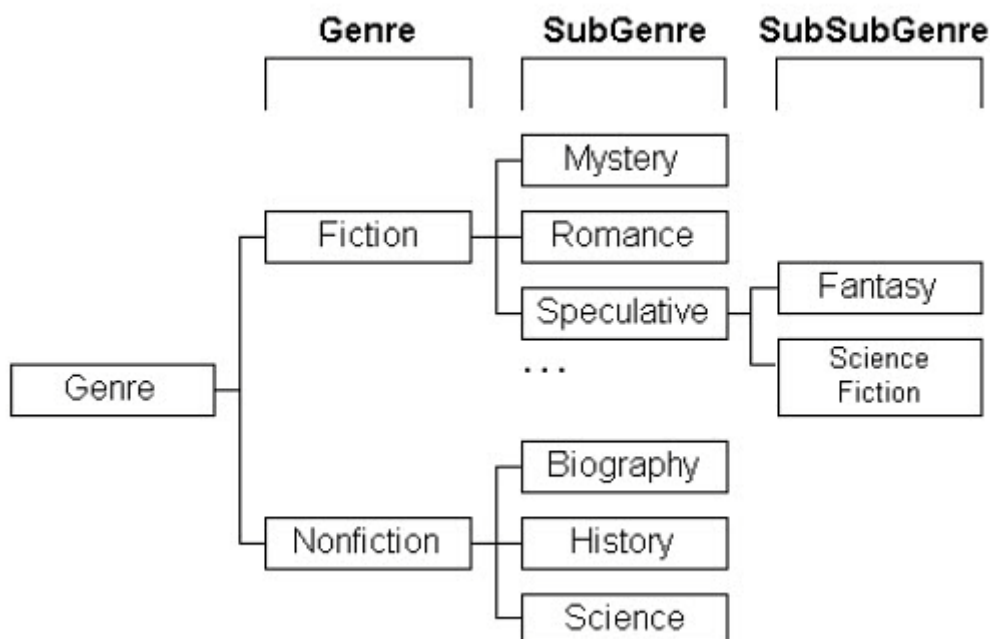
Genre  
SubGenre  
SubSubGenre

Values:

New Value:

☐ Show Advanced Settings

The generated taxonomy contains facets that correspond to the values of the various field names associated with the configured columns, for example:

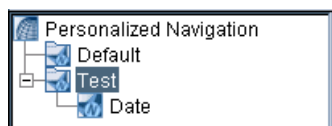




**Note:** The facets are ordered within the taxonomy in alphabetical order by default. You can change the order of the facets within a taxonomy using a comparator, as described in [The Personalized Navigation Application on page 101](#).

## Example: Creating a Taxonomy from Dates in a Database

In this example, we create a taxonomy from dates stored in a database using the structured column classifier. Assume that we have defined taxonomy called Date:



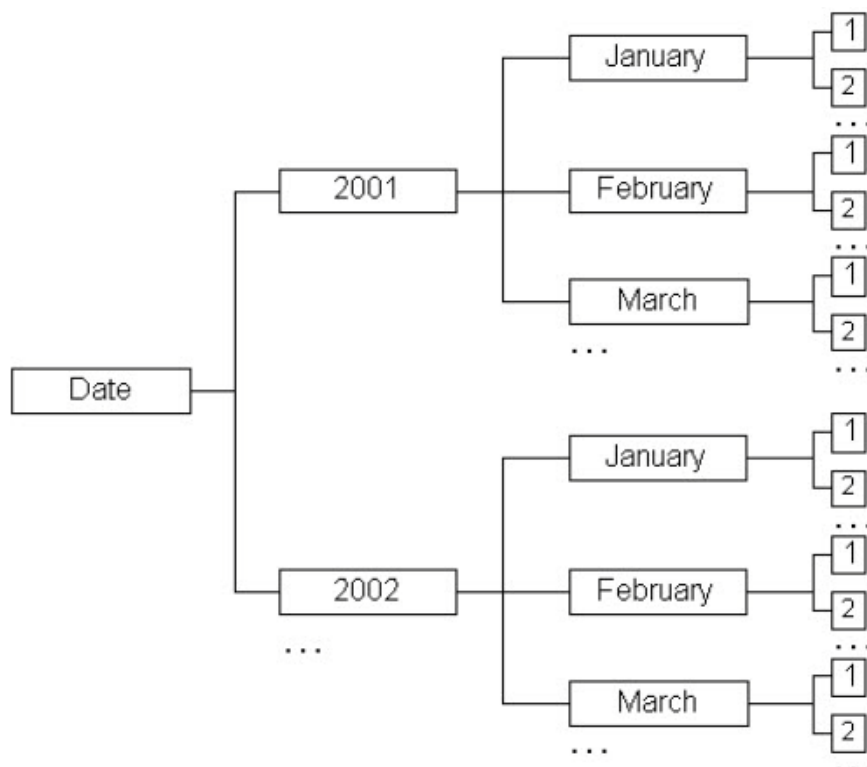
To configure the structured column classifier for the Date taxonomy:

- 1 Select **Date**, then select the **Classifiers** tab  
The Personalized Navigation application interface displays the Classifiers screen.
- 2 Select the **Structured Column** classifier type from the Structured Classifier group, and select **Add Classifier**  
The Structured Column classifier tab displays.
- 3 Configure the basic structured column classifier settings as follows:

Parameter	Configuration
<b>Datasource Name</b>	Select from the list of data sources configured for your application.
<b>Fields</b>	Use the New Value field to enter the name of the date column.

- 4 Select **Show Advanced Settings**, and select **Date Taxonomy Builder** from the Taxonomy Builder drop-down menu  
The Structured Column screen displays the Date Taxonomy Builder settings.
- 5 Accept the default (Create Standard) or configure additional Date classification options

The generated taxonomy contains standard date facets that correspond to the values of the date fields names associated with the configured columns, for example:



## Example: Creating a Price Taxonomy from a Database

In this example, we use the Structured Column classifier to create a taxonomy based on price.

To configure the Structured Column classifier for the Price taxonomy:

- 1 Create and select the facet named Price, then select the **Facet Settings** tab
- 2 Select the **Currency** comparator from the Advanced Facet Settings
- 3 Select the **Classifiers** tab, add the Structured Column classifier, and configure the basic settings as follows:

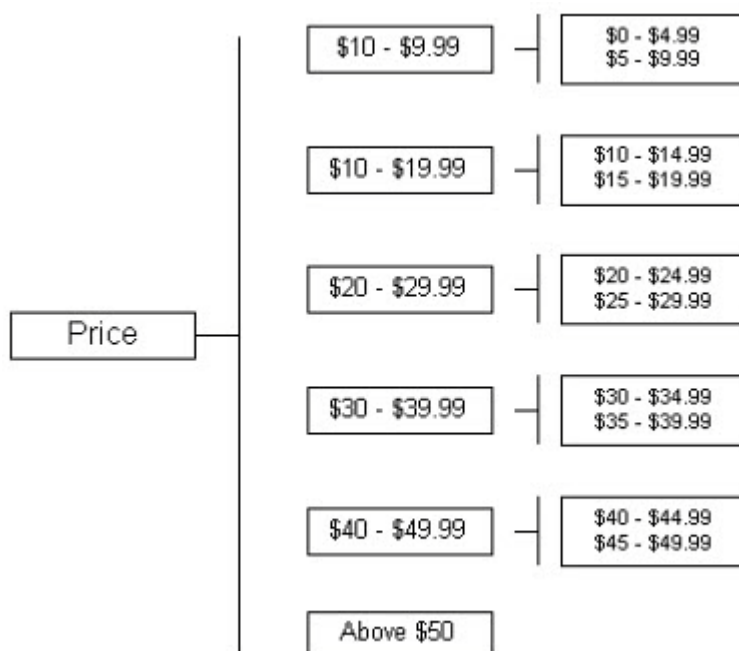
Parameter	Configuration
<b>Datasource Name</b>	Select the desired data source from the list of data sources configured for your application.
<b>Fields</b>	Use the New Value field to enter the name of the price column.

- 4 Select the **Range** taxonomy builder from the Advanced Classifiers setting, and specify the following values:

Parameter	Configuration
<b>Minimum Value</b>	Specify 0 to set the minimum price value to include in the facet. This value sets the lower boundary of the range of the facet.

<b>Maximum Value</b>	Specify 50 to set the maximum price value to include in the facet. This value sets the upper boundary of the range of the facet.
<b>Depth</b>	Specify 2 to set the number of levels of the taxonomy.
<b>Branching Factor</b>	Enter 5,2 to specify that the first level of the facet is divided into five equal partitions, and that the second level facets is divided into two.
<b>Tolerance</b>	Enter .01 to specify that values differing by this amount are treated as separate entries in the taxonomy, such that items priced at \$3.99 are selectable separately from item priced at \$3.98.

The resulting taxonomy contains the facets:



**Note:** There is no Below \$0 facet because no prices are below zero.

## Classifying Large Taxonomies by Accessing the Data at Runtime

You can use the Ad Hoc Column classifier to create a numeric range-based taxonomy, such as prices or years, dynamically at runtime when specifically selected by the user.

You should use the Ad Hoc Column classifier when:

- The results of conventionally classifying the data are very large.
- The performance loss of accessing the data source at runtime is acceptable.

You can also use the Ad Hoc Column classifier in combination with the Structured Column classifier to create a taxonomy that is partially pre-calculated and partially created at runtime.

**On-the-fly Range**

Description: <html>This classifier creates a numerical range based taxonomy, such as prices or years, at run time when specifically selected by a user. This should be done when the taxonomy would be too big if created before hand by the Structured Column classifier and the performance loss of continually accessing the data source is unimportant. <p> This classifier can also be used in combination with the Structured Column classifier to create a taxonomy that is partially precomputed and partially created at run time. </html>

On-the-fly Range -- Basic Settings

Datasource Name: CMS\_RESOURCES

Field name:

Minimum Value: 0.0 Maximum Value: 0.0

Branching Factor: 5 Tolerance: 0.0

You configure the Ad Hoc Column classifier by specifying the data source and column name that you want to classify. Specify minimum and maximum values for the specified field to include in the facet.

Specify a Branching Factor, which is the number of parts into which the specified range should be divided. The default value is 5, which specifies that a range of 0 - 50 is divided as:

```
0 - 10
10 - 20
20 - 30
30 - 40
40 - 50
```

Specify the Tolerance, which is maximum difference between two values that is significant. For example, for US currency, the tolerance is .01 (1 cent). The default is 0.

You can specify additional parameters to control the appearance of the generated facets, as described in [Configuring the Appearance of Structured Column Child Facet Names on page 141](#).

See [The Ad Hoc Column Classifier on page 157](#) for detailed descriptions of the Ad Hoc Column classifier settings.

## Classifying Content by Collection

The Collection classifier classifies all documents in the specified collections. You can use this classifier to define a taxonomy that matches all content in the specified collections at the document level.

When end users select a collection-based facet, the application restricts results to only documents from the associated collections.

**Collection**

Description: Classifies all documents in the specified collections.

Collection -- Basic Settings

Collections:

Values:

New Value: Add Delete

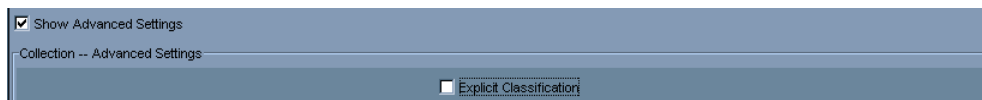
You configure the Collection classifier by specifying a list of the collections that you want to include in the facet definition. Specify collections by selecting from the list of collections configured for your application in the New Value field. The order in which you list the collections does not affect the facet.

**Note:** This classifier does not create child facets. There is no taxonomy builder associated with this classifier.

You can specify an advanced Collection facet setting to specify that document information for each matching document is stored explicitly in the application's index.

The Collection classifier matches every document in the specified collections; however by default it does not explicitly store the document information in the index; instead it uses a more economical method of associating the collection facet with the specified collection content.

There are some cases, such as when using Dictionary rules to perform advanced content restriction, in which the system requires that the document information is explicitly contained in the index.



You specify explicit classification using the Explicit Classification setting in the advanced settings. Select the checkbox to specify explicit classification.

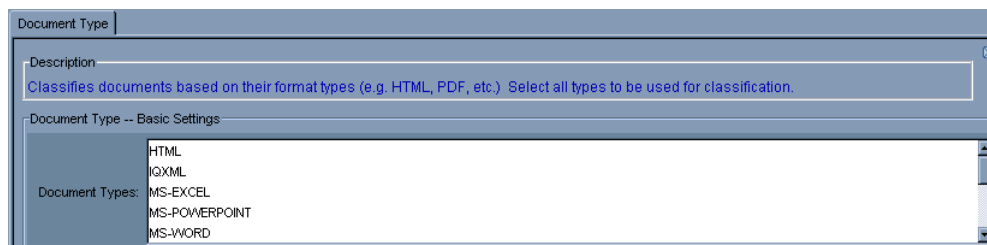
See [The Collection Classifier on page 159](#) for detailed descriptions of the Collection classifier settings.

## Classifying Content by Document Type

The Document Type classifier creates a taxonomy based on document type. It generates a facet for each selected document type:

- HTML
- IQXML
- MS-EXCEL
- MS-POWERPOINT
- MS-WORD
- PDF
- RTF
- TEXT
- UNKNOWN
- XML

When end users select a document type facet, the application restricts results to only documents of the specified types.

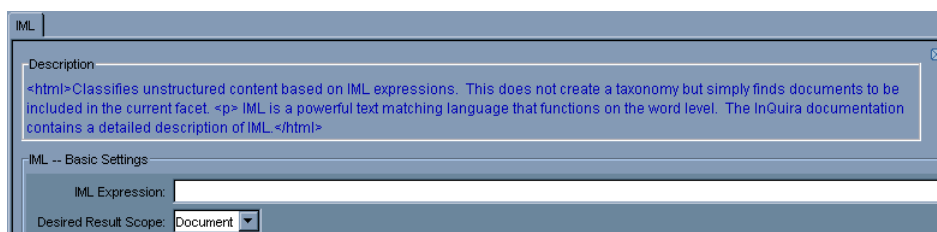


See [The Document Type Classifier on page 160](#) for detailed descriptions of the Document Type classifier settings.

## Using IML Expressions to Classify Content

The IML classifier classifies unstructured content based on InQuira Match Language (IML) expressions. See [Chapter 4, IML Reference](#) for more information on using IML.

You can use this classifier to define a facet that matches all content that matches the specified IML. When end users select an IML-based facet, the application restricts results to only content that matches the specified IML.



**Note:** This classifier does not create child facets. There is no taxonomy builder associated with this classifier.

You configure an IML classifier by specifying:

- The IML expression
- The scope you want to apply the expression to

For example, you could specify the expression `cat or dog` and the `Document` scope to match all documents that contain the words `cat` or `dog`.

Similarly, you could specify the `Title` scope to match only Titles that contain the words `cat` or `dog`.

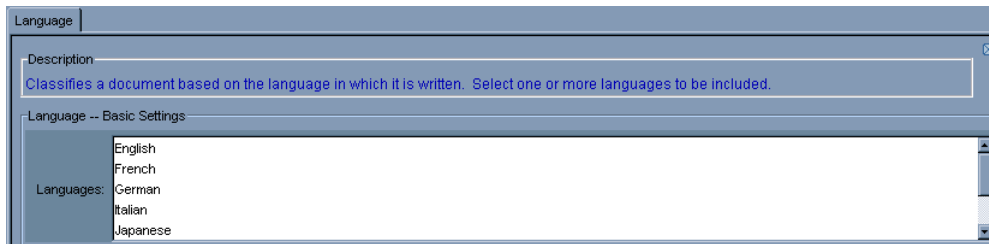
See [The IML Classifier on page 164](#) for detailed descriptions of the IML classifier settings.

## Classifying Content Based on Language

The Language classifier creates a taxonomy based on document language. It generates a facet for each selected document type:

- English
- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Traditional Chinese

When end users select a language-based facet, the application restricts results to documents identified as being in the specified languages.



This classifier can generate facets both for base languages (as listed above) and for localized languages, such as English:US or English:GB.

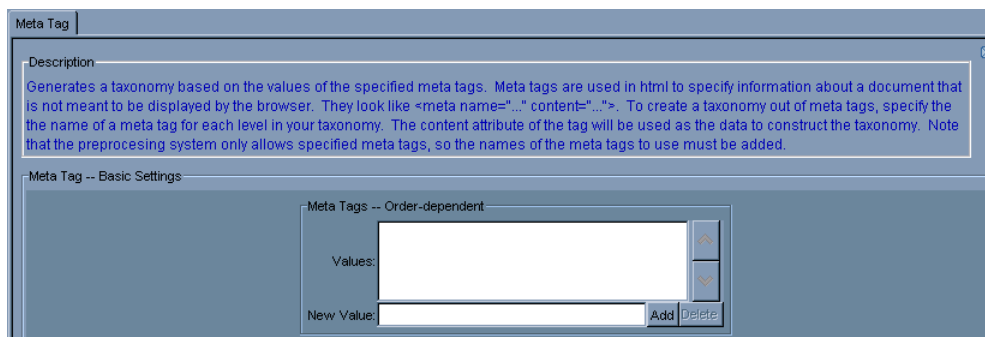
See [The Language Classifier on page 165](#) for detailed descriptions of the Language classifier settings.

## Classifying Content Based on HTML Meta Tags

The Meta Tag classifier generates a taxonomy based on the values of specified meta-tags within HTML documents. Meta-tags are mark-up that specify information about a document that is not meant to be displayed by the browser.

You can use this classifier to create multiple levels of facets based on the specified meta-tags of increasing specificity, for example:

```
<meta name="Genre" content="Fiction"/>
<meta name="SubGenre" content="Speculative"/>
<meta name="SubSubGenre" content="Fantasy"/>
```



You configure a meta-tag classifier by specifying the meta tag name for each level in your taxonomy. The Meta Tag classifier creates a facet for each unique content value within the specified tag, for example, each unique content value (*Fiction*, *Nonfiction*, etc.) associated with a *Genre* meta-tag. It creates a separate level for each meta-tag that you specify, in the order listed.

**Important!** You must ensure that the meta-tags that you specify for classification are added to the Preprocessor configuration file, as described in [Configuring the Preprocessor to Preserve Meta Tags for Classification on page 120](#).

You can use advanced Meta Tag classifier features to configure child facet creation by specifying and configuring a taxonomy builder.

See [Personalized Navigation Application Classification Components on page 155](#) for detailed descriptions of the Meta Tag classifier settings.

## Example: Creating a Taxonomy from Document Metadata

In this example, we classify documents based on the content of metadata tags using the Meta Tag classifier.

Consider a collection of documents that contain meta-data that define the following content categories:

```
<meta name="Genre" content="Fiction"/>
<meta name="SubGenre" content="Speculative"/>
<meta name="SubSubGenre" content="Fantasy"/>
```

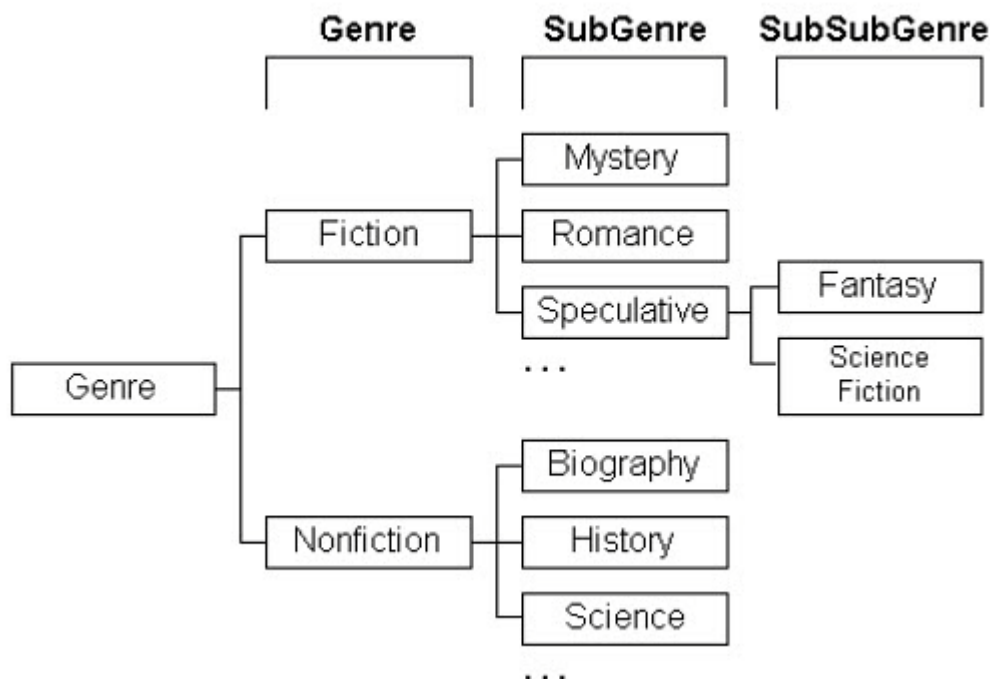
To configure the metadata classifier for the Genre facet:

- 1 Select the Genre facet, and select the Classifiers tab  
The Personalized Navigation interface displays the Classifiers screen.
- 2 Select the Meta Tag classifier type from the Unstructured Classifier group, and select Add Classifier  
The Meta Tag classifier tab displays.
- 3 Configure the meta tag classifier as follows:

Parameter	Configuration
Meta Tags	Use the New Value field to enter the field names in the order in which they occur in the taxonomy: <div>Genre</div> <div>SubGenre</div> <div>SubSubGenre</div>
Taxonomy Builder	Use the default String taxonomy builder. You configure the taxonomy builder in the Advanced Settings. The String taxonomy builder sorts sibling facets alphabetically by default. You can configure a comparator to sort the facets differently, as described in <a href="#">The Personalized Navigation Application on page 101</a> .



The generated taxonomy contains facets that correspond to the values of the various field names associated with the configured columns, for example:



## Classifying Content Based on Dictionary Concepts

The Ontology classifier generates a taxonomy from specified Dictionary concepts.

The Ontology classifier creates facets from the specified concepts, and creates additional levels of the taxonomy from the children of the specified concepts.

You can specify IML to restrict the content that matches the generated facets so that, for example, only documents that contain the facet label concept in the title match when the facet is selected.

**Ontology**

**Description**

<html>Generates a taxonomy from specified concepts in the ontology. Given a set of concepts to use as the start of taxonomies, this classifier will create taxonomies out of the concepts and their descendants. If you know exactly what concepts you want to use, add them to the "Include Concepts" list. The "%" symbol can be used as a wildcard when specifying concept names. <p> If you would like the system to find them for you, you can use the "Auto Detect Concepts." Specify representative concepts that you would like to be included, and the taxonomies will be found automatically. A good practice is to use the "%" wildcard to specify all customer-level concepts: e.g. noun.customername:%. <p> Finally, any concepts that should <i>not</i> be included in the taxonomies can be specified in the "Exclude Concepts" list.</html>

**Ontology -- Basic Settings**

**Auto Detect Concepts**

Values:

New Value:

**Include Concepts**

Values:

New Value:

**Exclude Concepts**

Values:

New Value:

**Note:** The preceding image only shows the top portion of the Ontology classifier display.

You use the Ontology classifier by specifying one or more entries to match:

- Concepts that you want the system to identify
- Concepts that you specify, either explicitly or by using the wildcard character

The Ontology classifier creates concept facet names in the form:

<concept\_description>

if the concept's description is short enough, or

<concept\_name>

without part of speech and domain information.

You specify IML to restrict the content match by specifying any valid IML, and selecting the desired result scope from the listbox.

See [The Ontology Classifier on page 161](#) for detailed descriptions of the Ontology classifier settings.

## Classifying Content by Directory Structure

The Site Graph classifier generates a taxonomy based on a specified directory structure. This produces a taxonomy that resembles a file explorer.

The screenshot shows the 'Site Graph -- Basic Settings' dialog box. At the top, there is a 'Description' field with a blue text box containing the text: 'Generates a taxonomy based on the directory structure of the documents. This will produce a taxonomy that looks much like a file explorer. Advanced settings are available to modify directory names, if desired.' Below this, there is a 'Collections' section with a 'Values' listbox and a 'New Value' dropdown menu. There are 'Add' and 'Delete' buttons next to the 'New Value' dropdown.

You specify the directory structures to classify by collection.

The screenshot shows the 'Site Graph -- Advanced Settings' dialog box. It has a 'Show Advanced Settings' checkbox which is checked. Below this, there is a 'Fetch URL' checkbox which is unchecked. The 'Directory Taxonomy Builder' section has a 'Prefixes' text field. Below this, there is another 'Show Advanced Settings' checkbox which is checked. The 'Directory Taxonomy Builder -- Advanced Settings' section contains several sub-sections: 'Combine Levels' with an unchecked checkbox, 'Comparators -- Order-dependent' with a 'Values' listbox containing 'Alphabetical' and a 'New Value' dropdown menu with 'Alphabetical', and 'Formatters -- Order-dependent' with a 'Values' listbox and a 'New Value' dropdown menu with 'Database Column Formatter'. There are also 'Regular expressions -- Order-dependent' and 'Regular expression replacements -- Order-dependent' sections, each with a 'Values' listbox and a 'New Value' text field. Each of these sections has 'Add' and 'Delete' buttons.

You can specify to classify content based on the URL of the document as it was crawled, rather than the URL that is displayed in the User Interface using the Fetch URL setting.

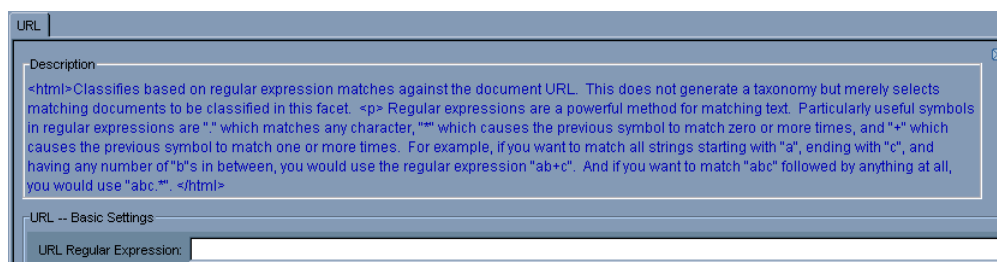
You can also modify directory names using the Prefix field to globally remove a specified prefix string from the URL, or by configuring the Directory taxonomy builder.

See [The Site Graph Classifier on page 162](#) for detailed descriptions of the Site Graph classifier settings.

## Classifying Content by URL

This classifier assigns content to the facet based on regular expression matches to document URLs.

**Note:** This classifier does not generate child facets. There is no taxonomy builder associated with this classifier.



You use the URL classifier by specifying a regular expression to match relevant characteristics of URL names within your application content.

**Note:** The URL classifier processes regular expressions according to the rules and standards defined for using regular expressions within Java applications.

Useful symbols in regular expressions include:

Symbol <sup>1</sup>	Function
.	matches any character
*	causes the previous character or expression to match zero or more times
+	causes the previous character or expression to match one or more times
(?i)	added to the beginning of a string, causes the following string to match regardless of case.

1. + indicates an obligatory match; \* indicates an optional match.

### Examples:

The regular expression:

"ab+c"

matches all strings that:

- start with "a"
- end with "c"
- and
- have any number of "b"s in between

The regular expression:

"abc.\*"

matches all strings that

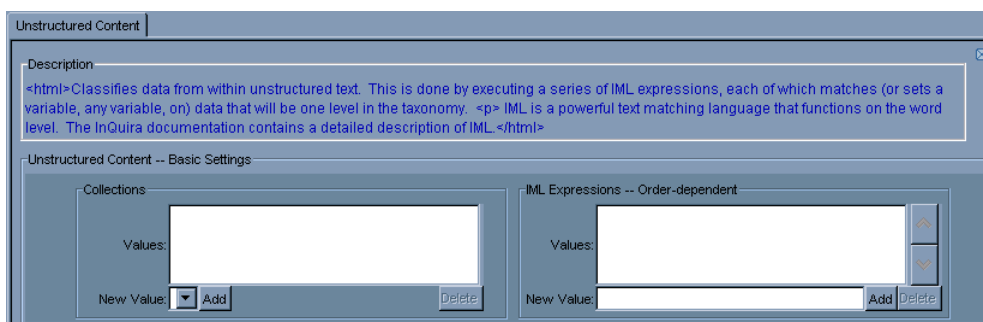
- start with "abc"
- and
- are followed by anything

See [The URL Classifier on page 163](#) for detailed descriptions of the URL classifier settings.

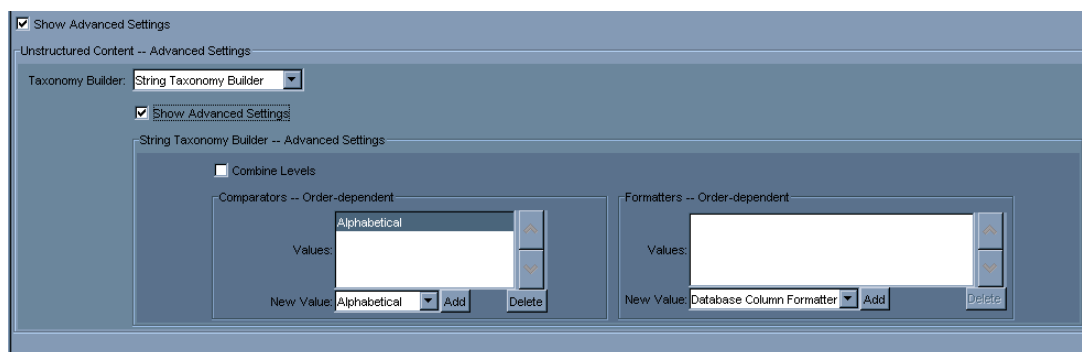
## Using the Unstructured Content Classifier

This classifier classifies data within unstructured content that matches specified IML expressions. Each IML expression matches (or sets a variable on) data that generates one level in the taxonomy.

See [Chapter 4, IML Reference](#) for more information about IML.



You use the Unstructured Content classifier by specifying one or more document collections, and one or more IML expressions to match.



You can also configure the taxonomy builder to determine child facet creation. The String taxonomy builder is the default.

See [The Unstructured Content Classifier on page 163](#) for detailed descriptions of the Unstructured Content classifier settings.

## Personalized Navigation Application Classification Components

The Personalized Navigation application provides access to the following content classification components:

- Structured data classifiers, as described in [Structured Data Classifiers on page 155](#)
- Unstructured data classifiers, as described in [Unstructured Data Classifiers on page 158](#)
- Structured or unstructured data classifier, as described in [Advanced Data Classifier on page 165](#)
- Taxonomy builders, as described in [Personalized Navigation Application Taxonomy Builders on page 169](#)
- Comparators, as described in [The Personalized Navigation Application on page 101](#)
- Formatters, as described in [Personalized Navigation Application Formatters on page 176](#)

### Structured Data Classifiers

Personalized Navigation provides the following classifiers for structured data.

Basic Classifiers	Description
<b>SQL Classifier</b>	This classifier selects all entries in the specified data source that match a specified SQL condition, as described in <a href="#">The SQL Classifier on page 155</a> .
<b>Structured Column Classifier</b>	This classifier creates a taxonomy based on the entries of one or more columns in a structured source, as described in <a href="#">The Structured Column Classifier on page 156</a> .

Advanced Classifiers	Description
<b>Ad Hoc Column Classifier</b>	This classifier creates a numerical range-based taxonomy, such as prices or years, at run time when specifically selected by a user, as described in <a href="#">The Ad Hoc Column Classifier on page 157</a> .

### The SQL Classifier

This classifier selects all entries in the specified data source that match a specified SQL condition.

**Note:** This classifier does not generate child facets. There is no taxonomy builder associated with this classifier.

Basic Setting	Description
<b>Datasource Name</b>	Select the desired data source from the structured data sources configured for your application.

<b>SQL Condition</b>	Specify an SQL condition that selects the appropriate values from the specified table for this facet. You use this classifier by specifying one or more conditions that include the contents of the <code>WHERE</code> clause, for example, <code>id=5</code> .
----------------------	---

## The Structured Column Classifier

This classifier creates taxonomies based on the entries of one or more columns in a structured datasource.

You can use this classifier to create multiple levels of facets of increasing specificity, for example, `Country -> State -> City`, or `Sector -> Industry -> Company`.

You can also use this classifier for structured data such as dates, prices, and URLs by selecting the appropriate taxonomy builder in the advanced settings.

The screenshot shows the 'Structured Column -- Basic Settings' dialog box. It has a 'Description' field with HTML text explaining the classifier's purpose. Below is the 'Structured Column -- Basic Settings' section, which includes a 'Datasource Name' dropdown menu set to 'CMS\_RESOURCES'. There is a 'Fields -- Order-dependent' list with a 'Values:' field and a 'New Value:' field. At the bottom right are 'Add' and 'Delete' buttons.

Basic Setting	Description
<b>Datasource Name</b>	Select the desired data source from the structured data sources configured for your application.
<b>Fields</b>	Specify one or more columns to use to create facets using the New Value field. The structured column classifier creates a separate level for each column that you specify.

The screenshot shows the 'Structured Column -- Advanced Settings' dialog box. It includes a 'Show Advanced Settings' checkbox at the top left. The main section is divided into two columns: 'Regular expressions -- Order-dependent' and 'Regular expression replacements -- Order-dependent'. Each column has a 'Values:' field and a 'New Value:' field. A tooltip 'You can drag an item to change its position within the list' is visible over the first 'Values:' field. At the bottom, there is an 'SQL View (Optional):' field and a 'Taxonomy Builder:' dropdown menu set to 'String Taxonomy Builder'. A 'Show Advanced Settings' checkbox is also at the bottom right.

Advanced Setting	Description
<b>Regular expressions</b>	Specify one or more optional regular expressions to match features of the column names that you want to replace. Use the New Value field to add regular expressions to the list. See <a href="#">Specifying Regular Expression Substitutions on page 175</a> for more information.
<b>Regular expression replacements</b>	Specify a corresponding replacement value for each regular expression. Use the New Value field to add replacement expressions. See <a href="#">Specifying Regular Expression Substitutions on page 175</a> for more information.
<b>SQL View</b>	Specify an optional view of the data source. This is generally required only when you need to perform a table join.
<b>Taxonomy Builder</b>	Select and configure an optional taxonomy builder to generate child facets for this facet, as described in <a href="#">Personalized Navigation Application Taxonomy Builders on page 169</a> . The default is to use the String taxonomy builder to create facets at all levels for which you have specified columns in the Fields setting.

## The Ad Hoc Column Classifier

You can use the Ad Hoc Column classifier to create a numeric range-based taxonomy, such as prices or years, dynamically at runtime when specifically selected by the user.

You should use the Ad Hoc Column classifier when:

- The results of conventionally classifying the data are very large.
- The performance loss of accessing the data source at runtime is acceptable.

You can also use the Ad Hoc Column classifier in combination with the Structured Column classifier to create a taxonomy that is partially pre-calculated and partially created at runtime.

## AD HOC COLUMN CLASSIFIER SETTINGS

Basic Setting	Description
<b>Datasource Name</b>	Select the desired data source from the structured data sources configured for your application.
<b>Field name</b>	Specify the column to use to create the facet.
<b>Minimum Value</b>	Specify the smallest value to include in the facet. The default is 0.0.
<b>Maximum Value</b>	Specify the largest value to include in the facet. The default is 0.0.

<b>Branching Factor</b>	<p>Specify the number of parts into which the specified range should be divided. The default value is 5, which specifies that the range of values is divided into five facets.</p> <p>For example, the range 0 – 50 is divided into:</p> <pre> 0 – 10 10 – 20 20 – 30 30 – 40 40 – 50 </pre>
<b>Tolerance</b>	<p>Specify the maximum difference between two values that is significant. For example, for US currency, setting the tolerance to .01 (1 cent), generates facets in the form \$10.00 - \$19.99.</p> <p>You can use the advanced Display Tolerance setting to round off facet values as described below.</p>

<b>Advanced Setting</b>	<b>Description</b>
<b>Display Tolerance</b>	Select this checkbox to subtract the Tolerance value from the range when displayed in the User Interface. For example, if the Tolerance value is set to .01 (1 cent), the range \$10.00 - \$15.00 is displayed as \$10.00 - \$14.99.
<b>Formatter</b>	Specify the formatter to generate the display name for the facet, as described in <a href="#">Personalized Navigation Application Formatters on page 176</a> . The default is the Database Column formatter, which displays the value of the column as the facet name.

## Unstructured Data Classifiers

Personalized Navigation provides the following basic and advanced classifiers for unstructured data.

<b>Basic Classifier</b>	<b>Description</b>
<b>Collection</b>	This classifier classifies all documents in the specified collections, as described in <a href="#">The Collection Classifier on page 159</a> .
<b>Document Type</b>	This classifier classifies documents based on their format types, for example HTML or PDF, as described in <a href="#">The Document Type Classifier on page 160</a> .
<b>Meta Tag Classifier</b>	This classifier generates a taxonomy based on the values of specified HTML meta-tags, as described in <a href="#">The Meta Tag Classifier on page 160</a> .



Basic Classifier	Description (Continued)
<b>Ontology Classifier</b>	This classifier generates a taxonomy from specified Dictionary concepts, as described in <a href="#">The Ontology Classifier on page 161</a> .
<b>Site Graph Classifier</b>	This classifier generates a taxonomy based on directory structure, as described in <a href="#">The Site Graph Classifier on page 162</a> .
<b>URL Classifier</b>	This classifier classifies based on regular expression matches against the document URL, as described in <a href="#">The URL Classifier on page 163</a> .
<b>Unstructured Content</b>	This classifier classifies data within unstructured content by executing a list of IML expressions, as described in <a href="#">The Unstructured Content Classifier on page 163</a> .
<b>IML Classifier</b>	This classifier classifies unstructured content based on IML expression matches, as described in <a href="#">The IML Classifier on page 164</a> .
<b>Language Classifier</b>	This classifier classifies a document based on the language in which it is written, as described in <a href="#">The Language Classifier on page 165</a> .

## The Collection Classifier

The Collection classifier classifies all documents in the specified collections. Only the facets for the collections are displayed.

**Note:** This classifier does not generate child facets. There is no taxonomy builder associated with this classifier.

Basic Setting	Description
<b>Collections</b>	Specify one or more collections to classify by selecting from the list of configured collections for the application.

Advanced Setting	Description
<b>Explicit Classification</b>	Select to specify that all documents be explicitly stored in the facet.  <b>Note:</b> This is only necessary to support advanced restriction of collections in the Dictionary rules. See <a href="#">Referring to Facets from Dictionary Rules on page 128</a> for more information.

## The Document Type Classifier

This classifier classifies documents based on their format types, such as HTML, PDF, or IQXML.

**Note:** This classifier does not generate child facets. There is no taxonomy builder associated with this classifier.

### Basic Setting

### Description

### Document Types

Select the document types that you want to classify from this list. Use the Shift and Ctrl keys to select multiple document types.

## The Meta Tag Classifier

This classifier generates a taxonomy based on the values of specified meta-tags.

### Basic Setting

### Description

### Meta Tags

Specify one or more meta tags to use to create facets using the New Values field. The Meta Tag classifier creates a new taxonomy level for each meta tag in the order specified.

Advanced Setting	Description
<b>Taxonomy Builder</b>	Select a taxonomy builder to use for this facet, as described in <a href="#">Personalized Navigation Application Taxonomy Builders on page 169</a> . The String Taxonomy Builder is the default.

## The Ontology Classifier

This classifier generates a taxonomy from specified Dictionary concepts and their children.

**Ontology**

**Description**

<html>Generates a taxonomy from specified concepts in the ontology. Given a set of concepts to use as the start of taxonomies, this classifier will create taxonomies out of the concepts and their descendants. If you know exactly what concepts you want to use, add them to the "Include Concepts" list. The "%" symbol can be used as a wildcard when specifying concept names.<p> If you would like the system to find them for you, you can use the "Auto Detect Concepts." Specify representative concepts that you would like to be included, and the taxonomies will be found automatically. A good practice is to use the "%" wildcard to specify all customer level concepts: e.g. noun.customername:%.<p> Finally, any concepts that should <i>not</i> be included in the taxonomies can be specified in the "Exclude Concepts" list.</html>

**Ontology -- Basic Settings**

**Auto Detect Concepts**

Values: [ ]

New Value: [ ] **Add** **Delete**

**Include Concepts**

Values: [ ]

New Value: [ ] **Add** **Delete**

**Exclude Concepts**

Values: [ ]

New Value: [ ] **Add** **Delete**

Basic Setting	Description
<b>Auto Detect Concepts</b>	Specify representative concepts for which you want the system to locate and generate taxonomies for similar concepts.
<b>Include Concepts</b>	Specify one or more concepts that you want to use as the basis of the taxonomy.
<b>Exclude Concepts</b>	Specify any concepts that you want to exclude from taxonomies. The Ontology classifier excludes the children of excluded concepts unless they are explicitly included as separate entries.

Advanced Setting	Description
<b>IML Expression</b>	Specify an optional IML expression to evaluate to determine which documents are matched by this facet. Use [] to designate the concept within the IML expression.
<b>Desired Result Scope</b>	The IML expression is searched for only within the specified scope <ul style="list-style-type: none"> <li>• Document</li> <li>• Title</li> <li>• Reference</li> <li>• Section</li> <li>• Subtitle</li> <li>• Sentence</li> <li>• Any</li> </ul>

## The Site Graph Classifier

This classifier generates a taxonomy based on a specified directory structure.

### Basic Setting

### Description

#### Collections

Select one or more collections to classify from the list of configured collections for the application. Use the New Value field to add collections to the list.

### Advanced Setting

### Description

#### Fetch URL

Select to classify based on the URL of the document as it was crawled, rather than the URL that is displayed in the User Interface.

#### Directory Taxonomy Builder

Specify settings for the Directory taxonomy builder, as described in [The Directory Taxonomy Builder on page 172](#).

## The URL Classifier

This classifier classifies based on regular expression matches against the document URL.

**Note:** This classifier does not generate child facets. There is no taxonomy builder associated with this classifier.

Setting	Description
URL Regular Expression	Specify one or more regular expressions to match the URLs.

Setting	Description
Fetch URL	Select to classify based on the URL of the document as it was crawled, rather than the URL that is displayed in the User Interface.

## The Unstructured Content Classifier

This classifier classifies data within unstructured content that matches specified IML expressions. Each IML expression matches (or sets a variable on) data that generates one level in the taxonomy.

See [Chapter 4, IML Reference](#) for more information about IML.

Basic Setting	Description
Collections	Specify one or more collections use for this facet.
IML Expressions	Specify one or more IML expressions to evaluate to determine which documents are matched by this facet. Each IML expression generates a new level in the taxonomy.

Advanced Setting	Description
Taxonomy Builder	Specify settings for a taxonomy builder, as described in <a href="#">Personalized Navigation Application Taxonomy Builders on page 169</a> . The String taxonomy builder is the default.

## The IML Classifier

This classifier classifies unstructured content based on matching specified IML expressions.

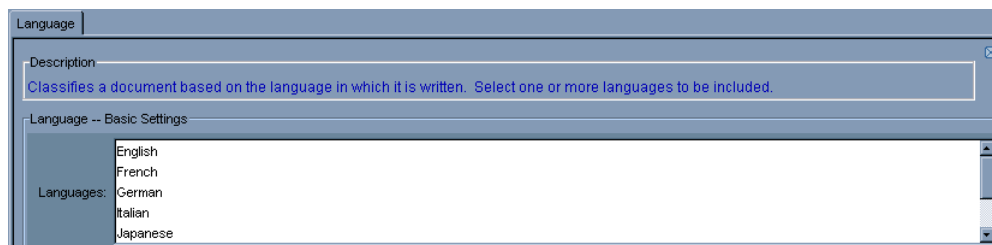
Refer to the [Chapter 4, IML Reference](#) for more information on IML.

**Note:** This classifier does not generate any child facets. There is no taxonomy builder associated with this classifier.

Setting	Description
IML Expression	Specify an IML expression to evaluate to determine which documents are matched by this facet.
Desired Result Scope	Select the scope to restrict matching of the IML expression: <ul style="list-style-type: none"> <li>• Document</li> <li>• Title</li> <li>• Reference</li> <li>• Section</li> <li>• Subtitle</li> <li>• Sentence</li> <li>• Any</li> </ul>

## The Language Classifier

This classifier classifies documents based on their encoded language.



Basic Setting	Description
Language	Select one or more languages to include in this taxonomy. Each selected language creates an separate facet.

## Advanced Data Classifier

Personalized Navigation provides the Template advanced classifier for structured or unstructured data.

### The Template Classifier

The Template classifier adds further classification to an existing taxonomy. For every element of the taxonomy, this classifier attempts to apply the template and, if the template successfully creates a new classifier, it is added to that element.

This is particularly useful in creating a taxonomy for both unstructured and structured data. The taxonomy could, for example, be initially created with the Structured Column classifier and then a Template classifier could be used to add IML classifiers for the unstructured data.

To access the Template classifier:

- 1 On the Classifiers tab, select **Classifier Group: Other**.
- 2 Select **Advanced Classifiers**.  
**Template** displays in the **Classifier Type** field.
- 3 Click **Add Classifier** to define the Template classifier.



The Personalized Navigation window displays the Template Settings section.

Basic Setting	Description
<b>Template</b>	<p>Select a classifier template to use for this facet. The options are:</p> <ul style="list-style-type: none"> <li>• IML</li> <li>• On-the-fly Range</li> <li>• Structured To Unstructured</li> <li>• URL</li> </ul> <p>Additional settings must be defined for each type of template. See the following sections for details.</p>

## IML TEMPLATE SETTINGS

Classifies unstructured content based on IML expressions. This does not create a taxonomy but simply finds documents to be included in the current facet.

IML is a powerful text matching language that functions on the word level. The Oracle Knowledge documentation contains a detailed description of IML.

Basic Setting	Description
<b>IML Expression</b>	Specify an IML expression to evaluate to determine which documents are matched by this facet.
<b>Desired Result Scope</b>	<p>Select the scope to restrict matching of the IML expression:</p> <ul style="list-style-type: none"> <li>• Document</li> <li>• Title</li> <li>• Reference</li> <li>• Section</li> <li>• Subtitle</li> <li>• Sentence</li> <li>• Any</li> </ul>
<b>Collections</b>	Select one or more collections to classify from the list of configured collections. Use the New Value field to add collections to the list.

## ON-THE-FLY RANGE TEMPLATE SETTINGS

This classifier creates a numerical range based taxonomy, such as prices or years, at run time when specifically selected by a user. This should be done when the taxonomy would be too big if created prior to



run time by the Structured Column classifier and the performance loss of continually accessing the data source is unimportant.

This classifier can also be used in combination with the Structured Column classifier to create a taxonomy that is partially precomputed and partially created at run time.

Basic Setting	Description
<b>Datasource Name</b>	Select the desired data source from the structured data sources configured for your application.
<b>Field Name</b>	Specify the column to use to create the facet.
<b>Display Tolerance</b>	Select to subtract the Tolerance value from the range when displayed in the User Interface. For example, if the Tolerance value is set to .01 (1 cent), the range \$10.00 - \$15.00 is displayed as \$10.00 - \$14.99.  See <a href="#">Tolerance</a> below for more information.
<b>Minimum/Maximum Value</b>	Specify the smallest and largest value to include in the facet. The default is 0.0.
<b>Branching Factor</b>	Specify the number of parts into which the specified range should be divided. The default value is 5, which specifies that the range of values are divided into five facets. For example, the range 0 - 50 is divided into:  0 - 10 10 - 20 20 - 30 30 - 40 40 - 50
<b>Tolerance</b>	Specify the maximum difference between two values that is significant. For example, for US currency, setting the tolerance to .01 (1 cent), generates facets in the form \$10.00 - \$19.99.  Use the <a href="#">Display Tolerance</a> setting to round off facet values in the UI.
<b>Formatter</b>	Specify the formatter to generate the display name for the facet, as described in <a href="#">Personalized Navigation Application Formatters on page 176</a> . The default is the <a href="#">Database Column Formatter</a> , which displays the value of the column as the facet name.

## STRUCTURED TO UNSTRUCTURED TEMPLATE SETTINGS

### Basic Setting

### Description

#### Bridge Column

Specify the column to use to create the facet.

#### Template

Options are:

- IML – See [IML Template Settings on page 166](#).
- URL – See [URL Template Settings on page 168](#).

## URL TEMPLATE SETTINGS

Classifies based on regular expression matches against the document URL. This does not generate a taxonomy but merely selects matching documents to be classified in this facet.

Regular expressions are a powerful method for matching text. Three particularly useful symbols in regular expressions are:

### Symbol      Description

- (Dot) In the default mode, this matches any character except a newline.
- \* (Asterisk) Causes the previous symbol to match zero or more times. *ab\** matches *a*, *ab*, or *a* followed by any number of *b*'s.
- + (Plus) Causes the previous symbol to match one or more times. *ab+* will match 'a' followed by any non-zero number of 'b's; it will not match just 'a'.

For example, if you want to match all strings starting with *a*, ending with *c*, and containing any number of *bs* in between, you would use the regular expression *ab+c*. If you want to match *abc* followed by anything at all, you would use *abc\**.

Basic Setting	Description
<b>URL Regular Expression</b>	Specify one or more regular expressions to match the URLs.
<b>Case Sensitive</b>	Select to indicate that the URL expression is case sensitive.
<b>Fetch URL</b>	Select to classify based on the URL of the document as it was crawled, rather than the URL that is displayed in the User Interface.

## Personalized Navigation Application Taxonomy Builders

Taxonomy Builders are procedures that you configure within classifiers to create new facets during the classification process. Taxonomy builders enable you to:

- Generate facets in your taxonomy programmatically
- Automatically update facets within data repositories that are subject to frequent change

Taxonomy builders provide a convenient method of automating the classification process, which in many cases can be difficult or impossible to perform manually.

For example, to classify a database having `category` and `subcategory` fields, we could create each facet by hand, and configure a structured column classifier to select the appropriate row for each facet. However, this be an impractical approach for data containing hundreds of facets, especially if the structure is subject to frequent change.

Taxonomy builders use the data from the classifier in which they are configured to construct new taxonomy sub-facets. Each time the data is classified, the classifier reconstructs the taxonomy to correspond to any changes to the referenced data.

You can use taxonomy builders with the following classifiers:

- The Structured Column classifier
- The Meta Tag classifier
- The Site Graph classifier (only the Directory taxonomy builder is available)

You configure taxonomy builders by specifying values for various parameters that correspond to properties of the data that you want to classify. You can configure taxonomy builders to classify data based on directory structures, dates, text strings, and ranges of values, as described in:

- The String Taxonomy Builder
- The Date Taxonomy Builder
- The Directory Taxonomy Builder
- The Range Taxonomy Builder

## The String Taxonomy Builder

The String taxonomy builder generates taxonomy entries based on the text strings contained within the target data.

You configure the String taxonomy builder by specifying various optional and required parameters. The Directory Taxonomy Builder screen displays the configuration settings:

The following table describes the String Taxonomy Builder settings.

Advanced Parameter	Description
<b>Combine Levels</b>	Select this parameter to specify to combine any non-branching paths into a single facet.
<b>Comparators</b>	Use this field to specify optional comparators to determine the order of the facets. See <a href="#">The Personalized Navigation Application on page 101</a> for more information on using Comparators within Taxonomy Builders.
<b>Formatters</b>	Use this field to specify optional formatters to determine the labels of facets within the User Interface. See <a href="#">Personalized Navigation Application Formatters on page 176</a> for more information on using Formatters within Taxonomy Builders.

## The Date Taxonomy Builder

The Date taxonomy builder generates taxonomy entries based on date values within the target data. The Date taxonomy builder also contains settings to create facets for dynamically defined dates, such as Today and This Week.

You configure the Date taxonomy builder by specifying various optional and required parameters. The Date Taxonomy Builder screen displays the basic and advanced configuration settings:

Basic Parameters	Description
<b>Create Standard</b>	Use this field to specify a basic taxonomy that includes years, months, and days.
<b>Create Today</b>	Use this field to create a facet for items occurring today. For example, to categorize news articles by their posting date. <b>Note:</b> This facet requires classification on a daily basis.
<b>Create This Week</b>	Use this field to create a facet for items occurring this week. Weeks are defined as Sunday - Saturday. <b>Note:</b> This facet requires classification on a daily basis.
<b>Create This Month</b>	Use this field to create a facet for items occurring this month. <b>Note:</b> This facet requires classification on a daily basis.
<b>Create This Year</b>	Use this field to create a facet for items occurring this year. For example, to categorize books by their publication date. <b>Note:</b> This facet requires classification on a daily basis.
<b>Create All Dates</b>	Use this field to create a facet containing all dates.
<b>Combine Levels</b>	Select this parameter to specify to combine any non-branching paths into a single facet.
<b>Today Label</b>	Use this field to specify the text for the Today facet, for example, Today's Articles.
<b>This Week Label</b>	Use this field to specify the text for the This Week facet.
<b>This Month Label</b>	Use this field to specify the text for the This Month facet.
<b>This Year Label</b>	Use this field to specify the text for the This Year facet.
<b>All Dates Label</b>	Use this field to specify the text for the All Dates facet.

<b>Language</b>	Use this field to specify the language in which the dates should be rendered. Supported languages include any that java supports.
<b>Comparators</b>	Use this field to specify optional comparators to determine the order of the facets. See <a href="#">The Personalized Navigation Application on page 101</a> for more information on using Comparators within Taxonomy Builders.
<b>Formatters</b>	Use this field to specify optional formatters to determine the presentation of facets within the User Interface. See <a href="#">Personalized Navigation Application Formatters on page 176</a> for more information on using Formatters within Taxonomy Builders.

## The Directory Taxonomy Builder

The Directory taxonomy builder generates taxonomy entries based on directory paths within a file system. The result resembles a file browser.

You configure the Directory taxonomy builder by specifying various optional and required parameters. The Directory Taxonomy Builder screen displays the basic and advanced configuration settings:

Basic Parameter	Description
<b>Prefixes</b>	Use this optional parameter to remove a specified common prefix in the directory names from the facet names.  <b>Note:</b> To remove multiple prefixes, use the Regular Expressions field in the Advanced section.
Advanced Parameter	Description
<b>Combine Levels</b>	Select this parameter to specify to combine any non-branching paths into a single facet.
<b>Comparators</b>	Use this field to specify optional comparators to determine the order of the facets. See <a href="#">The Personalized Navigation Application on page 101</a> for more information on using Comparators within Taxonomy Builders.

<b>Formatters</b>	Use this field to specify optional formatters to determine the presentation of facets within the User Interface. See <a href="#">Personalized Navigation Application Formatters on page 176</a> for more information on using Formatters within Taxonomy Builders.
<b>Regular Expressions</b>	Use this field to specify optional regular expressions to transform matching strings to the corresponding expressions specified in the Regular Expressions replacements field. See <a href="#">Specifying Regular Expression Substitutions on page 175</a> for more information on using regular expression substitution within Taxonomy Builders.
<b>Regular expression replacements</b>	Use this field to specify optional regular expression replacements for strings that match the corresponding expressions specified in the Regular Expressions field. See <a href="#">Specifying Regular Expression Substitutions on page 175</a> for more information on using regular expression substitution within Taxonomy Builders.

## The Range Taxonomy Builder

You can use the Range taxonomy builder to generate taxonomy entries that subdivide a range of values. The resulting taxonomy is a subdivided range of values. The Range Taxonomy Builder screen displays following basic and advanced settings.

The screenshot shows the 'Range Taxonomy Builder' interface. At the top, a dropdown menu is set to 'Range Taxonomy Builder'. Below it, the 'Basic Settings' section includes checkboxes for 'Do Above' and 'Do Below', both of which are checked. There are input fields for 'Minimum Value' (0.0), 'Maximum Value' (0.0), 'Depth' (2), and 'Tolerance' (0.0). A 'Branching Factors -- Order-dependent' section contains a list with the value '5', a 'Values' input field, and a 'New Value' input field with 'Add' and 'Delete' buttons. Below the basic settings is a checkbox for 'Show Advanced Settings' which is checked. The 'Advanced Settings' section includes a checkbox for 'Display Tolerance', input fields for '"Above" Label' (Over) and '"Below" Label' (Under), and a 'Formatter' dropdown menu set to 'Database Column Formatter'.

Basic Parameter	Description
<b>Do Above</b>	Use this field to create a facet containing all items whose value is greater than the Maximum Value setting, for example Over \$50.00.
<b>Do Below</b>	Use this field to create a facet containing all items whose value is less than the Minimum Value setting, for example Under \$5.00.
<b>Minimum Value</b>	Use this field to specify the minimum value to use as a range boundary, for example, \$5.00.
<b>Maximum Value</b>	Use this field to specify the maximum value to use as a range boundary, for example, \$50.00.
<b>Depth</b>	Use this field to specify the number of levels for the taxonomy. For example, specify a depth of 2 to create top level facets and second level facets such that selecting the top level facet displays the second level facets.

<b>Tolerance</b>	Use this field to specify the maximum difference between two values that is significant, such as .01 (one cent) for currency.
<b>Branching Factors</b>	Use this field to specify a list of values to divide the taxonomy ranges into. For example, enter 5, 2 to specify that the top level taxonomy is divided into five facets, and the second level is divided into two facets. If there are more levels (specified by the Depth parameter) than branching entries, the lower levels use the last entry in the list.
<b>Advanced Parameter</b>	<b>Description</b>
<b>Display Tolerance</b>	Use this field to specify a value to subtract from the range when displaying, so that the range \$10-\$15 displays as \$10-\$14.99.
<b>Above Label</b>	<p>Use this field to specify the name of the facet showing values more than the Maximum. You can use the characters [] which are replaced by the maximum value. For example, specify:</p> <p>Over []</p> <p>to display</p> <p>Over \$50.00</p> <p>where 50 is the value of the Maximum parameter.</p>
<b>Below Label</b>	<p>Use this field to specify the name of the facet showing values less than the Minimum. You can use the characters [] which are replaced by the minimum value. For example, specify:</p> <p>Under []</p> <p>to display</p> <p>Under \$50.00</p> <p>where 50 is the value of the Minimum parameter.</p>
<b>Formatters</b>	Use this field to specify optional formatters to determine the presentation of facets within the User Interface. See <a href="#">Personalized Navigation Application Formatters on page 176</a> for more information on using Formatters within Taxonomy Builders.



## Specifying Regular Expression Substitutions

The Structured Column classifier and the Directory taxonomy builder provide fields in which you can specify regular expressions and corresponding replacement expressions to capture and replace strings to use as facet names.

You specify regular expression replacements as two separate lists of corresponding values.

Parameter	Description
<b>Regular Expression</b>	Specify one or more regular expressions to match some characteristic of the field or directory name.  Each entry must be paired with a corresponding replacement entry.
<b>Regular Expression replacements</b>	Specify one or more strings as replacements for the characteristics of the field or directory names that match the specified regular expressions.

Replacements can include grouping syntax. For example, specifying:

```
([0-9]+)
```

and the replacement expression

```
"$1"
```

results in all numbers being quoted.

## Personalized Navigation Application Comparators

Comparators are modules that compare two values and determine whether the first is less than, equal, or greater than the second. You use Comparators to determine the order in which the facets are displayed in the User Interface. You use Comparators within the String, Date, and Directory taxonomy builders.

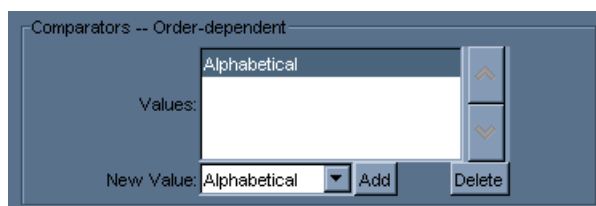
You specify Comparators as a list of one or more items. Each entry in the list applies to the corresponding level of the taxonomy. If there are more levels than comparator entries, the lower levels use the last entry in the list.

For example, to specify a top level of a taxonomy level sorted alphabetically, followed by a second level sorted by date, enter:

```
Alphabetical
Number
```

The resulting taxonomy might resemble:

```
Fiction
  Mystery
  Romance
  Speculative
Nonfiction
  Biography
  History
```



The Personalized Navigation application provides the following comparators:

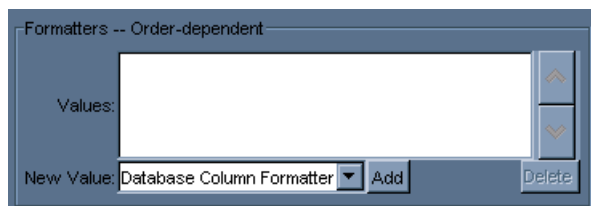
Comparator	Description
<b>Alphabetical</b>	Use this comparator to specify that facets be sorted alphabetically.
<b>Number</b>	Use this comparator to specify that facets be sorted numerically.
<b>Currency</b>	Use this comparator to specify that facets be sorted by currency in ascending order.
<b>Dimension</b>	Use this comparator to sort facets based on dimensions that include regular numbers (1, 3, 29, etc.), fractions (1/4, 3/8, etc.), and fractions combined with numbers (e.g. 1-1/4, 1 1/4, 19-3/8, etc.).
<b>Last Name</b>	Use this comparator to specify that facets be sorted alphabetically by last name.
<b>Month</b>	Use this comparator to specify that facets be sorted in calendar order by month name.

## Personalized Navigation Application Formatters

Formatters are procedures that perform string substitution on facet names as they are created by taxonomy builders, so that you can control their presentation within the User Interface.

You use Formatters within the String, Date, Directory, and Range taxonomy builders.

You specify Formatters as a list of one or more items. Each entry in the list applies to the corresponding level of the taxonomy. If there are more levels than formatter entries, the lower levels use the last entry in the list.



The Personalized Navigation application provides the following formatters:

Formatter	Description
<b>Database Column Formatter</b>	Use this formatter to specify that the User Interface displays the value of the configured column as closely as possible to the corresponding value on the answer page.
<b>Name Formatter</b>	Use this formatter to specify that the User Interface displays person name values in the form <surname>, <name>.

# User Manager

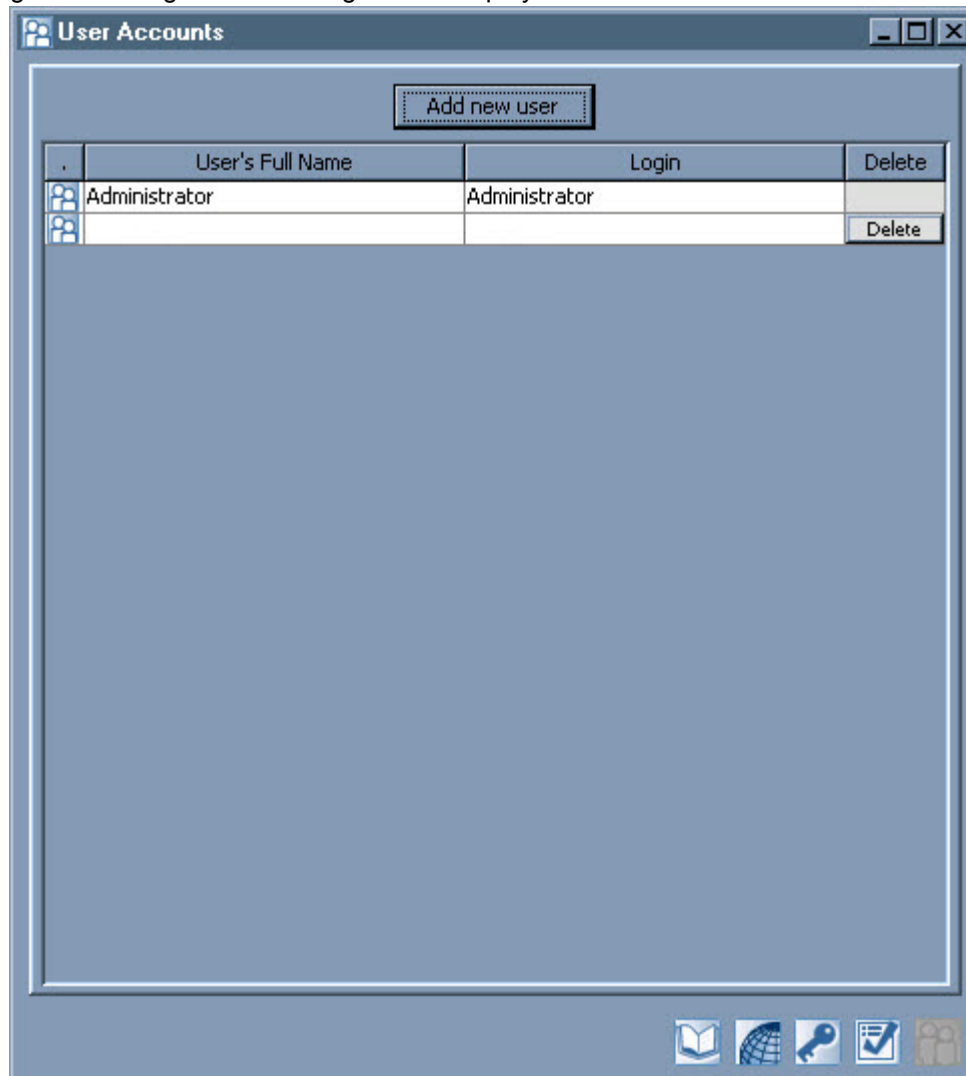
The Oracle Knowledge User Manager is a component accessed from within the Language Workbench that allows you to create and set permissions for Language Workbench users. To access User Manager please see [Intelligent Search Language Tuning Guide](#).

This Chapter explains:

- [Using User Manager on page 178](#)
- [Add New User on page 179](#)

## Using User Manager

Upon opening User Manager the following screen displays:



The following installed Users appear in addition to any custom created user accounts.:

**User Login**  
**Administrator**

**Default Password**  
This is the password setup

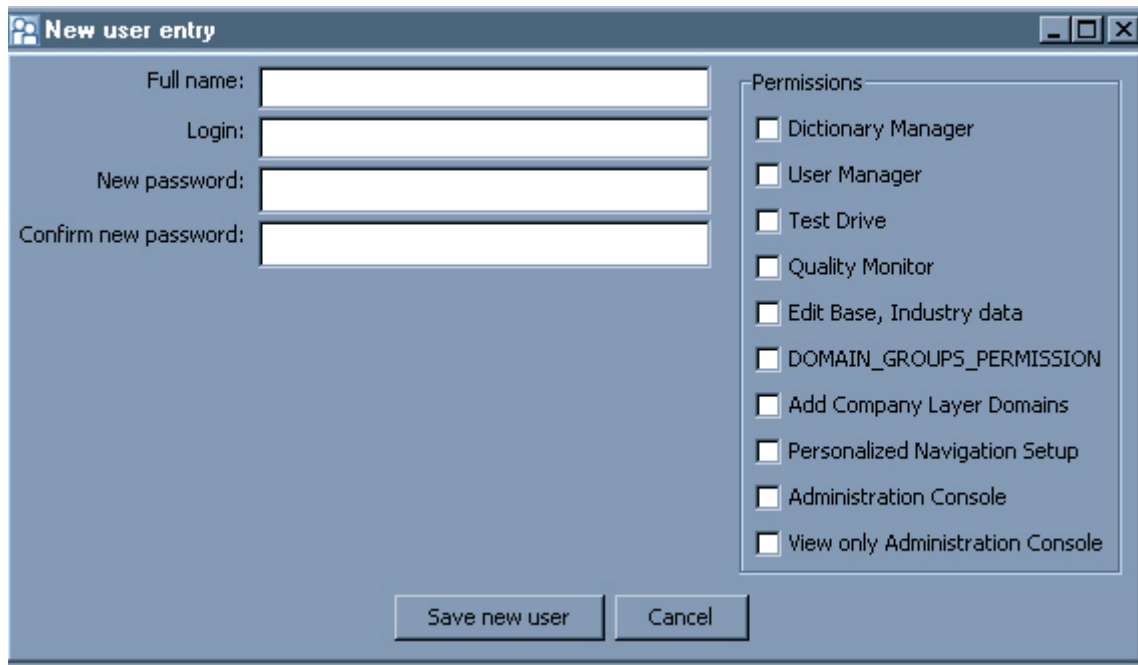
**Permissions**

- Dictionary Manager
- User Manager
- Test Drive
- Quality Monitor
- Edit Base, Industry data
- DOMAIN\_GROUPS PERMISSIONS
- Add Company Layer Domains
- Personalized Navigation Setup
- Administration Console

## Add New User

You can add Language Workbench Users by selecting **Add new user** from the User Manager screen. The New user entry screen displays:

**Note:** Only Language Workbench Users with the User Manager permission set are able to create and modify user accounts



- 1 Enter the name of the User in the **Full name** field.
- 2 Enter the name to be used for logging into the Language Workbench in the **Login** field.
- 3 Enter the password to be used with the Login in the **New password** field.  
**Note:** The password must be at least eight characters long, and contain both letters and numbers.
- 4 Reenter the password in the **Confirm new password** field.
- 5 Select the Permissions to be associated with the User. For more information, please see Permissions.
- 6 Click **Save new user** to save the newly created user. Upon saving, the Username appears in the User Accounts screen.

## User Permissions

The following are the permissions available to each user:

<b>Dictionary Manager</b>	Allows the User to access Dictionary Manager. You can assign the following advanced Dictionary Manager permissions described below: <ul style="list-style-type: none"> <li>• Edit Base, Industry Data</li> <li>• Add Company Layer Domains</li> </ul>
<b>User Manager</b>	Allows the User to access User Manager.
<b>Test Drive</b>	Allows the User to access Test Drive.
<b>Quality Monitor</b>	Allows the User to access Quality Monitor.
<b>Edit Base, Industry data</b>	Allows the User to create and modify Dictionary Objects that belong to Base and Industry level domains in Dictionary Manager.
<b>DOMAIN_GROUPS PERMISSIONS</b>	Allows the User to create and modify Domains and Domain Lists used in Dictionary Manager.
<b>Add Company Layer Domains</b>	Allows the User to create Company Layer Domains used in Dictionary Manager.
<b>Personalized Navigation Setup</b>	Allows the User to access Personalized Navigation Setup.
<b>Administration Console</b>	Allows the User to access System Manager to configure Oracle Knowledge Intelligent Search.
<b>View only Administration Console</b>	Allows the User to view but not modify configurations in System Manager.

## Modify a User Account

To modify a User Account:

- 1 Double-click the Username you want to modify from the User Accounts screen.
- 2 After you make modifications, select **Save new user**. The changes take effect automatically.

## Delete a User Account

To delete a user account: highlight the username and select **Delete**. The user name deletes.

# Configuring Access to Business Data

Oracle Knowledge accesses business data through the Preference Service, which accesses and stores business data items that it obtains from configured data sources. The data sources that the Preference Service accesses are called *providers*, and the data items that it uses are called *preferences*.

You can configure Oracle Knowledge to access business data from the following types of Providers:

- The current Oracle Knowledge session
- External sources, such as CRM packages

You access business data for use by the Business Rules module, which is a separate component that enables you to:

- Configure the Preference Service to access Preference data from Providers
- Set Business Conditions within Rules based on user, session, and other context information

Business Conditions within Rules enable you to tailor application response based on various aspects of:

- User identity and activity
- Session activity

The Rules Engine evaluates Business Conditions within Rules by comparing the values obtained by the Preference Service to the values specified within the Business Condition.

## The Preference Service

The Preference Service manages the business data preferences obtained by the configured data providers.

Oracle Knowledge includes a session provider that is installed and configured with the application. The session provider is designed to access only preference data for a given session; it does not store persistent preference data for use in multiple sessions. See [The Session Provider on page 182](#) for more information.

## Access to Preference Data

The Preference Service uses a key to obtain preference values from configured providers. Keys take the form of multiple strings concatenated by the period ( . ) character, for example:

```
string.string.string
```

The concatenated parts enable namespace traversal for data retrieval, as described in [Hierarchical Namespace Traversal on page 182](#).

## Hierarchical Namespace Traversal

The multi-part key format that the Preference Service uses to access preference data supports hierarchical name space traversal for data retrieval.

The Preference Service locates the value of keys by attempting to match a complete key name, for example, `color.car.ext`. If it fails to locate the required key, it can create and use progressively smaller sub-keys, for example `color.car`, then `color`.

You can configure a preference to require exact key matching rather than hierarchical name space traversal using the `exactMatch` parameter, as described in [Configuring Preferences on page 185](#).

## Preference Types

You can configure Oracle Knowledge to access the following types of business data:

- **User context data:** User context data includes user profile information, such as user name and IP address.
- **Session context data:** Session context data includes session-dependent information, such as URLs and pages visited.
- **Question context data:** Question context enables the Rules Engine to dynamically set the value of a configured preference based on information in the user's question.

You can access data using Oracle Knowledge's configured session provider, or using a custom provider that you configure using the process described in [Creating Custom Preferences on page 184](#).

[Session Provider Preferences on page 183](#) describes the user and session context preferences that are configured for the Oracle Knowledge Session Provider.

## Business Data Providers

Providers are the means of accessing various sources of business data, which can include the web- or application server hosting the Oracle Knowledge session, as well as external sources of business data, such as CRM applications.

The Oracle Knowledge session provider is installed and configured with the product. [Session Provider Preferences on page 183](#) describes the individual preferences that are configured for the session provider.

You can define and configure additional providers using the process described in [Creating Custom Preferences on page 184](#).

The Context Variables page in the Advanced Configuration Facility allows selection of the following providers, in addition to Session: Request Attribute, Request Header, and Request Parameter.

## The Session Provider

Oracle Knowledge includes a session provider that is installed and configured with the application. The session provider is designed to access only preference data for a given session; it does not store persistent preference data for use in multiple sessions.



[Session Provider Preferences on page 183](#) describes the individual preferences that are configured for these providers.

## Session Provider Preferences

The Session provider is installed and configured with the Oracle Knowledge product. The Session provider defines the following Preferences that you can include in business conditions:

Preference	Description
<b>Request Base URL</b> <b>Request Referer Page</b> <b>User Host</b> <b>User IPAddress</b> <b>User Name</b>	The Preference Service obtains user data from a Principal object ( <code>javax.security.Principal</code> ) in the request. The principal object can be created by: <ul style="list-style-type: none"><li>• a user agent, as in JSP and ASP pages</li><li>• the HTTP request headers, as are present for standard login message boxes in web browsers</li><li>• a login request</li><li>• a custom preference handler, as described in <a href="#">Creating Custom Preferences on page 184</a></li></ul>
<b>Time (Now)</b> <b>Today's Date</b>	The Preference Service obtains time and date data from the current system time. This data is not stored in the Preference Service.

## Creating Custom Providers

You can define and configure custom Providers for the application by:

- Creating a custom provider that implements the Provider application programming interface (API)
- Defining the custom provider to the application, as described in [Defining a Provider in the Application on page 184](#)
- Configuring Preferences to use the provider, as described in [Configuring Preferences on page 185](#)

## Defining a Provider in the Application

You define a Provider using the providers map in the application configuration file. The providers map includes an entry for each provider defined for the application. Providers that are defined in the map are then available to individual preferences, as described in [Configuring Preferences on page 185](#).

The following example shows the default Session provider configuration in the providers map:

```
<map name="providers" type="PreferenceProvider" />
<entry name="sessionProvider" type="PreferenceProvider" />
```

where:

<b>providers</b>	Specifies the map that defines the Preference providers. You must have at least one session provider entry in this list. The Session provider is defined by default for the application.
<b>sessionProvider</b>	Is an entry that refers to the default sessionProvider. Each defined provider must be specified as an entry in this map.

## Creating Custom Preferences

You can define and configure custom business data preferences for the application by:

- Creating a custom Preference that implements the Preference application programming interface (API), as described in [The Provider API on page 193](#).
- Adding the Preference to the application, as described in [Configuring Preferences on page 185](#).

## Configuring Preferences

You can configure Preferences for the Preference Service using the Advanced Configuration Facility. The Preferences page of the Advanced Configuration Facility accesses a preferences map within the application configuration file that specifies parameters for each defined Preference.

The following example shows the format of the preferences map:

```
<map name="preferences"
  controller="com.inquiria.preference.PreferenceController">
  <entry name="key" type="string" />
  <entry name="traversalType" type="integer" />
  <entry name="exactMatch" type="boolean" />
  <list name="providers" type="PreferenceProvider" />
  <entry name="purpose" type="string" />
  <entry name="dataTypeClass" type="string" />
</map>
```

The Preference parameter elements correspond to the used Preference parameters in the Advanced Configuration Facility, as described in [Adding and Modifying Context Variable Preference Parameters on page 187](#).

where:

<b>Key</b>	Specifies the key required to access this Preference, for example, <code>color.car.ext</code>
<b>Traversal Type</b>	<p>Specifies how to traverse the hierarchical name space for keys that contain multiple parts when multiple providers are defined for this preference. Valid values are:</p> <p style="text-align: center;"><code>Names First</code></p> <p>Exhaust the name space in the first provider before asking the next provider</p> <p style="text-align: center;"><code>Providers First</code></p> <p>Try the full key on all providers before processing sub-keys</p>
<b>Exact Match</b>	<p>Is a Boolean value that specifies whether the full key is required to find the Preference. Valid values are:</p> <ul style="list-style-type: none"> <li>• On- the full key is required</li> <li>• Off - the full key is not required</li> </ul>
<b>Purpose</b>	Specifies whether the Preference is accessible from the Business Rules module. Specify <code>BusinessCondition</code> to make the Preference available to the Business Rules module.
<b>Data Type Class Name</b>	<p>Specifies the data type for this Preference, as defined in the Business Condition module. Valid values are:</p> <ul style="list-style-type: none"> <li>• Date</li> <li>• Facet</li> <li>• Integer</li> <li>• Real</li> <li>• String</li> <li>• Time</li> </ul>

<b>Modifiable by Search System</b>	Specifies that the value of the parameter can be changed dynamically by the Business Condition module. See <a href="#">Chapter A, Configuring Access to Business Data</a> for more information on creating Business Conditions that set Preference values based on question context information.
<b>Providers</b>	Specifies the Providers that the application can use to access data for this Preference. You must specify at least one provider to configure a Preference. The Preference Service processes the list of providers in the order specified.

## Editing Preferences

Create and modify preferences using the Context Variables section of the Advanced Configuration Facility.

To edit preferences, select **Context Variables** from the Advanced Configuration Facility menu.

## The Context Variables Page

The Context Variables page displays the currently defined context variables available to the Business Condition module. Each context variable corresponds to a configured preference that has a defined purpose of `BusinessCondition`.

## Displaying the Editable Context Variable List

To display context variables as selectable items, select the **Edit** button on the Context Variables page.

**ORACLE** Advanced Configuration [HELP ?](#)

**Editing: Context Variables**

**OK** **Cancel**

Variables :

<a href="#">date:today</a>		
<a href="#">facet:selected</a>		
<a href="#">formevent.COMMON_USER_SESSION_ID</a>		
<a href="#">formevent.DEVICE_ID</a>		
<a href="#">formevent.FIELD_ID</a>		
<a href="#">formevent.FORM_ID</a>		
<a href="#">formevent.FORM_PAGE_INSTANCE_ID</a>		
<a href="#">formevent.FORM_VERSION</a>		
<a href="#">formevent.PAGE_ID</a>		
<a href="#">formevent.SECTION_ID</a>		
<a href="#">formevent.SESSION_ID</a>		
<a href="#">request.baseURL</a>		
<a href="#">request.referrerURL</a>		
<a href="#">result_language</a>		
<a href="#">site</a>		
<a href="#">time:now</a>		
<a href="#">user.address</a>		
<a href="#">user.host</a>		
<a href="#">user.name</a>		
<a href="#">user.segment</a>		

[Add New Item](#)

**OK** **Cancel**

The Editing:Context Variables page displays the list of defined context variables as hypertext links, and displays an Add New Item link. You can select an existing context variable to modify its preference criteria, or define a new context variable.

## Selecting a Context Variable for Editing

You can select an existing context variable to modify its preference criteria, or define a new context variable.

To modify an existing preference, select the corresponding context variable from the list.

To define a new preference and context variable, select **Add New Item** from the list.

ORACLE<sup>®</sup> Advanced Configuration [HELP ?](#)

**Editing: Context Variables > Variables**

**OK** **Cancel**

Item Name ▶ formevent.PAGE\_ID

**Variables**

Key ▶ formevent.PAGE\_ID

Traversal Type ▶ Names First

Exact Match : ☒ On ☐ Off

Purpose : Business Condition

Data Type Class Name : String

Modifiable by Search System : ☐ On ☒ Off

Providers : 1. Providers : Request Parameter

[Add New Item](#)

**OK** **Cancel**

The Editing:Context Variables:Variables page displays fields that define the preference for the selected variable.

## Adding and Modifying Context Variable Preference Parameters

You can use the fields on the Editing:Context Variables:Variables page to add or modify preference parameters for the selected context variable.

- 1 Specify values for the parameters, as described in [Configuring Preferences on page 185](#).
- 2 Click **OK** to update the application configuration.

## The Preference API

You can create a preference handler by implementing the Preference API. The Preference API is defined in `com.inquiria.preference.PreferenceService`. It includes the following methods:

- [The `getPreference` Method on page 188](#)
- [The `getExternalString` Method on page 188](#)
- [The `getExternalCode` Method on page 189](#)
- [The `getPreferenceValue` Method on page 189](#)
- [The `SetPreferenceValue` Methods on page 189](#)
- [The `rundown` Method on page 191](#)
- [The `getPreferences` Method on page 192](#)
- [The `getPreferenceKeys` Method on page 192](#)

### The `getPreference` Method

<b>Method:</b>	<b><code>getPreference</code></b>
<b>Syntax:</b>	<pre>public Preference getPreference(java.lang.String key)</pre>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>Description:</b>	This method retrieves the preference definition for a key from the configuration service. If the preference has not been defined, a dynamic preference object is created and returned.
<b>Returns:</b>	The Preference object that corresponds to the requested key

### The `getExternalString` Method

<b>Method:</b>	<b><code>getExternalString</code></b>
<b>Syntax:</b>	<pre>public java.lang.String getExternalString(java.lang.String key)</pre>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>Description:</b>	This method returns the string representation of the preference value as defined by an external provider.
<b>Returns:</b>	Returns the string value for the specified preference

## The getExternalCode Method

<b>Method:</b>	<b>getExternalCode</b>
<b>Syntax:</b>	<pre>public java.io.Serializable getExternalCode(java.lang.String key)</pre>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>Description:</b>	This method returns the native form of the preference value as defined by the provider. This can be null if no key access using this preference value is required that is based not on the external string.
<b>Returns:</b>	The code value for specified preference

## The getPreferenceValue Method

<b>Method:</b>	<b>getPreferenceValue</b>
<b>Syntax:</b>	<pre>public PreferenceValue getPreferenceValue(java.lang.String key)</pre>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>Description:</b>	This method returns the <code>PreferenceValue</code> object that corresponds to this key.
<b>Returns:</b>	The preference value

## The SetPreferenceValue Methods

There are several `setPreferenceValue` methods within the Preference API. The various `setPreferenceValue` methods use various combinations of key, value, code, string, and preference parameters.

<b>Method:</b>	<b>setPreferenceValue</b>
<b>Syntax:</b>	<pre>public PreferenceValue setPreferenceValue(java.lang.String key, java.lang.String value)</pre>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>value</b>	Specifies the string value
<b>Description:</b>	This method returns the <code>PreferenceValue</code> object that corresponds to this key.
<b>Returns:</b>	The updated <code>PreferenceValue</code> object

<b>Method:</b>	<b>setPreferenceValue</b>
<b>Syntax:</b>	<pre>public PreferenceValue setPreferenceValue(java.lang.String key, java.io.Serializable code)</pre>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>Description:</b>	This method sets the code for the preference to the <code>code</code> argument, the string value to <code>code.toString()</code> with the given key, and returns the updated <code>PreferenceValue</code> object. If no preference with this key exists, it creates a new preference using only the session preferences.
<b>Returns:</b>	The updated <code>PreferenceValue</code> object

<b>Method:</b>	<b>setPreferenceValue</b>
<b>Full Name:</b>	<pre>public PreferenceValue setPreferenceValue(Preference preference, java.lang.String value)</pre>
<i>where:</i>	
<b>key</b>	Specifies the key value
<b>value</b>	Specifies the string value
<b>Description:</b>	This method sets the value for the preference with the given key and returns the updated <code>PreferenceValue</code> object. If no preference with this key exists, it creates a new preference using only the session preferences.
<b>Returns:</b>	The updated <code>PreferenceValue</code> object

<b>Method:</b>	<b>setPreferenceValue</b>
<b>Syntax:</b>	<pre>public PreferenceValue setPreferenceValue(Preference preference, java.lang.String value, java.io.Serializable code)</pre>
<i>where:</i>	
<b>preference</b>	Specifies the preference
<b>value</b>	Specifies the string value
<b>code</b>	Specifies the code value
<b>Description:</b>	This method sets the value and code for the preference with the given preference and returns the updated <code>PreferenceValue</code> object. If no preference with this key exists, it creates a new preference using only the session preferences.
<b>Returns:</b>	The updated <code>PreferenceValue</code> object

<b>Method:</b>	<b>setPreferenceValue</b>
----------------	---------------------------



<b>Syntax:</b>	<code>public PreferenceValue setPreferenceValue(java.lang.String key, java.lang.String value, java.io.Serializable code)</code>
<i>where:</i>	
<b>value</b>	Specifies the string value
<b>code</b>	Specifies the code value
<b>Description:</b>	This method sets the value and code for the preference with the given key and returns the updated <code>PreferenceValue</code> object. If no preference with this key exists, it creates a new preference using only the session preferences.
<b>Returns:</b>	The updated <code>PreferenceValue</code> object

<b>Method:</b>	<b>setPreferenceValue</b>
<b>Syntax:</b>	<code>public PreferenceValue setPreferenceValue(PreferenceValue value)</code>
<i>where:</i>	
<b>value</b>	Specifies the preference value
<b>Description:</b>	This method sets the value for the preference with the given key and returns the passed <code>PreferenceValue</code> object. If no preference with this key exists, it creates a new preference using only the session preferences.
<b>Returns:</b>	The updated <code>PreferenceValue</code> object

## The rundown Method

<b>Method:</b>	<b>rundown</b>
<b>Syntax:</b>	<code>public void rundown() throws TransportException</code>
<b>Description:</b>	The rundown method is called after every completion of a request from the <code>Execution.rundown()</code> method.
<b>Throws:</b>	<code>TransportException</code>

Refer to `rundown()` in the JavaDoc documentation for more information.

## The getPreferences Method

<b>Method:</b>	<b>getPreferences</b>
<b>Syntax:</b>	<code>public java.util.Collection getPreferences()</code>
<b>Description:</b>	This method retrieve a list of all defined preferences. The returned list contains <code>Preference</code> objects.
<b>Returns:</b>	A list of all <code>Preference</code> objects.

## The getPreferenceKeys Method

<b>Method:</b>	<b>getPreferenceKeys</b>
<b>Syntax:</b>	<code>public java.util.Collection getPreferenceKeys()</code>
<b>Description:</b>	This method retrieve a list of all defined preference keys. The returned list contains <code>String</code> objects.
<b>Returns:</b>	A list of all <code>String</code> objects

## The Provider API

You can create a custom provider to access business data by implementing the Provider API. Providers must extend the class `com.inquirapreference.AbstractProvider`, which includes the following methods:

- [The Provider Default Constructor on page 193](#)
- [The `getType` Method on page 193](#)
- [The `get` Methods on page 194](#)
- [The `set` Method on page 195](#)
- [The `toString` Method on page 195](#)
- [The Provider `runDown` Method on page 195](#)

### The Provider Default Constructor

<b>Method:</b>	<b>AbstractProvider</b>
<b>Syntax:</b>	<code>public AbstractProvider()</code>
<b>Description:</b>	Specifies the default constructor. A default constructor is required so that the class can be loaded dynamically.

### The `getType` Method

<b>Method:</b>	<b>getType</b>
<b>Description:</b>	This method should not be overridden. The static <code>String</code> variable type should be initialized in a static initializing code block. For example: <pre>static {type = 'my type'}</pre>
<b>Returns:</b>	The provider type as a string

## The get Methods

There are two `get` methods within the Provider API; a namespace hierarchy traversal method, and a non-traversal method.

<b>Name:</b>	<b>get</b>
<b>Syntax:</b>	<code>public PreferenceValue get(java.lang.String key, byte depth, byte traversalDepth)</code>
<i>where:</i>	
<b>key</b>	Specifies the preference name
<b>depth</b>	Specifies the number of elements in the key
<b>traversalDepth</b>	Specifies the number of elements that can be removed from the key to create sub-keys
<b>Description:</b>	<p>This method retrieves the value that corresponds to the key argument within this provider. The number of key elements is specified by the <code>depth</code> parameter.</p> <p>This function traverses the namespace hierarchy for the key by attempting to locate sub-keys if the full key location fails. Sub-keys are created by removing the final key element; for example:</p> <pre>color.car.ext -&gt; color.car -&gt; color</pre> <p>You can limit namespace traversal by specifying a traversal depth argument. This argument specifies the number of key elements that can be removed to create sub-keys. For example:</p> <pre>key = color.car.ext depth = 3 traversalDepth = 1</pre> <p>specifies that the only valid keys are <code>color.car.ext</code> and <code>color.car</code>.</p> <p><code>get</code> is dependent on the <code>get(String)</code> method, which must be implemented by all subclasses. This method does not have to be overridden in custom providers unless the functionality is not required.</p>
<b>Returns:</b>	<p>a <code>PreferenceValue</code> object with the appropriate values set</p> <p><code>null</code> if no match is found</p>

<b>Method:</b>	<b>get</b>
<b>Syntax:</b>	<code>protected abstract PreferenceValue get(java.lang.String key)</code>
<i>where:</i>	
<b>key</b>	Specifies an exact key name

<b>Description:</b>	This abstract method is called by <code>get(String,byte,byte)</code> to find the value for a given key. This method does not create and match sub-keys to traverse the key namespace; it only checks for an exact key match. This method must be defined by all provider classes and must implement the specific functionality for this provider.
<b>Returns:</b>	<ul style="list-style-type: none"> <li>the <code>PreferenceValue</code> object, if it locates the exact key</li> <li><code>null</code>, if it does not locate the exact key</li> </ul>

## The set Method

<b>Method:</b>	<b>set</b>
<b>Syntax:</b>	<pre>protected abstract boolean set(java.lang.String key, PreferenceValue value)</pre>
<i>where:</i>	
<b>value</b>	Specifies the preference value
<b>Description:</b>	This abstract method sets the value for the implied preference of the value object.
<b>Returns:</b>	The preference value

## The toString Method

<b>Method:</b>	<b>toString</b>
<b>Syntax:</b>	<pre>public java.lang.String toString()</pre>
<b>Description:</b>	This method returns information about the provider that can be useful for debugging.
<b>Overrides:</b>	<code>toString</code> in the class <code>java.lang.Object</code>
<b>Returns:</b>	Information about the provider

## The Provider rundown Method

<b>Method:</b>	<b>rundown</b>
<b>Syntax:</b>	<pre>public abstract void rundown()</pre>
<b>Description:</b>	This method is called when the preference services' <code>rundown</code> method is called. This method should store and retain preference values for this provider.