ORACLE
PEOPLESOFT ENTERPRISE

PeopleTools 8.51 PeopleBook: Feed Publishing Framework

August 2010

ORACLE

PeopleTools 8.51 PeopleBook: Feed Publishing Framework
SKU pt8.51tfeed-b0810

Copyright © 1988, 2010, Oracle and/or its affiliates. All rights reserved.

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions Warranty/Consequential Damages Disclaimer

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.

Warranty Disclaimer

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.

Restricted Rights Notice

If this software or related documentation 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 USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software 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

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 8

Chapter 9

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Publishing Framework Preface

This preface discusses:

- PeopleTools modules.
- PeopleBooks and the online PeopleSoft library.

PeopleTools Modules

This PeopleBook refers to the following modules:

- Feed Publishing Framework.
- Integration Broker.
- PeopleSoft Query.
- Workflow.
- Process Scheduler.

Feed Publishing Framework

The Feed Publishing Framework is a module of Oracle's PeopleTools. It consists of unified interfaces for content owners and administrative users to create, configure, and maintain feed definitions of various types; interfaces for end users to discover related feeds and search feed definitions; a set of APIs and code samples to assist application developers in creating new type of feeds and integrating them with existing features; and multiple language support.

Integration Broker

Integration Broker facilitates exposing PeopleSoft business logic as services and consuming external web services by Oracle's PeopleSoft applications. Integration Broker also supports synchronous and asynchronous messaging among PeopleSoft applications and with third-party systems. Integration Broker uses a variety of communication protocols, while managing message structure, message content, and transport disparities.

PeopleSoft Query

Oracle's PeopleSoft Query is an end user reporting tool. With Query Manager, you can extract the precise information that you are looking for by using visual representations of your PeopleSoft database, without writing Structured Query Language (SQL) statements. The queries that you write can be as simple or as complex as necessary; they can be one-time queries or queries that you use repeatedly.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Workflow

Workflow enables you to efficiently automate the flow of information throughout your enterprise, crossing both application and functional boundaries. PeopleSoft Workflow technology consists of a powerful set of tools that enables you to automate time-consuming business processes and deliver the right information to the right people at the right time. You can merge the activities of multiple users into flexible business processes to increase efficiency, cut costs, and keep up with rapidly changing customer and competitive challenges.

Process Scheduler

Process Scheduler is a centralized tool that enables application developers, system administrators, and application users to manage PeopleSoft batch processes. Using the PeopleSoft Pure Internet Architecture, you can access a list of processes through a web browser and queue and run a process request.

PeopleBooks and the PeopleSoft Online Library

A companion PeopleBook called *PeopleBooks and the PeopleSoft Online Library* contains general information, including:

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.
- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.
- How to manage the locally installed PeopleSoft online library, including web site folders.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Application abbreviations found in application fields.

You can find this companion PeopleBook in your PeopleSoft online library.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 1

Getting Started With the Feed Publishing Framework

This chapter provides an overview of Feed Publishing Framework and discusses its implementation.

Feed Publishing Framework Overview

The Feed Publishing Framework provides user interfaces and APIs for feed definition creation and maintenance, feed searching and subscription, as well as feed document generation and delivery.

Feeds

A web feed, news feed, or feed is a data format that makes rapidly changing content available to users. A feed contains entries, which might be headlines, full-text articles, excerpts, summaries, digital media, and links to content on a web site, along with various metadata. PeopleSoft applications, like news agencies, external web sites, and other content distributors publish or syndicate web feeds, to which users can subscribe. In PeopleSoft applications some examples of feeds include worklists, queries, discussions, and so on.

Web feeds work by using the pull technology model. Typically, when using web feeds, a content provider publishes a feed link on their site which you register in an aggregator program (also called a feed reader or a news reader) that runs on your own machines. Feed readers can be independent programs or browser extensions that provide configurable integrated feed reader functionality. Web-based feed readers seldom require additional software installation and make the user's feeds available on any computer with internet access. Microsoft Outlook and Mozilla Thunderbird are two commonly used secure feed readers.

When directed, the aggregator polls all the servers in its feed list to ascertain if new content exists; if so, the aggregator retrieves the new content. You can configure feed readers to check for new or updated content at regular intervals. When you no longer want to receive updated content, you remove the feed from the reader.

Atom and Really Simple Syndication (RSS) are the two primary syndication formats. Atom also provides a standardized way to export an entire blog, or parts of it, for backup or for importing into other blogging systems. Some web sites let people choose between RSS- or Atom-formatted web feeds; others offer only RSS or only Atom. In particular, many blog and wiki sites offer their web feeds in the Atom format.

Most web pages, including PeopleSoft application pages, use this feed icon to indicate that a feed is available in either the RSS 1.0, RSS 2.0, and Atom 1.0 formats.

Note. PeopleSoft applications offer feeds in Atom 1.0 format only.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Publishing Framework

The Feed Publishing Framework provides:

- A set of APIs and code samples to assist application developers in creating new type of feeds and integrating them with existing features.
- Unified interfaces for content owners and administrative users to create, configure, and maintain feed definitions of various types.
- Interfaces for end users to discover related feeds and search feed definitions.

With the Feed Publishing Framework, you can develop and publish feeds from any PeopleSoft or non-PeopleSoft data. Then, with the delivered user interfaces, users can find and subscribe to your feeds.

Feed Publishing Framework Implementation

This section lists the prerequisites for the Feed Publishing Framework implementation and discusses the steps to:

- Use delivered feed data types.
- Develop new feed data types.

Prerequisites

The Feed Publishing Framework relies on other PeopleTools components—such as Integration Broker—for you to be able to publish and consume feeds.

Your PeopleSoft system must be configured as follows:

Step	*Reference*
The Integration Broker gateway and target nodes must be configured and active.	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring the Integration Broker Gateway and Target Nodes, page 19.
Integration Broker service target locations must be configured.	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring Integration Broker Service Target Locations, page 23. **Note.** If you plan to use secure authentication, then you must also configure SSL on your system. See *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Working with Oracle WebLogic," Implementing WebLogic SSL Keys and Certificates and *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Working with IBM WebSphere," Setting Up SSL For WebSphere.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Step	*Reference*
The default user of the ANONYMOUS mode must be a valid user.	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring the Default User of the ANONYMOUS Node, page 24.
The default local node must have authentication set to password or certificate. In addition, the content URI text and portal URI text of the default local node must be defined.	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring the Default Local Node, page 25.
The content URI text and portal URI text of each local host node must be defined.	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Setting URI Text for Local Host Nodes, page 27.
For Oracle WebLogic Server, you must configure the server to disable its own authentication.	**Note.** By default, the delivered config.xml file is set to disable Oracle WebLogic Server's own authentication. No additional configuration is required unless you have changed this authentication setting. Disabling Oracle WebLogic Server's authentication allows authentication to be passed through and handled by the PeopleSoft servlet. See Appendix B, "Disabling Authentication on Oracle WebLogic Server," page 119.
Assign users the roles and permission necessary to use feeds. Feed administrators require the PTPT1300 – Portal Administrator permission list, which is included in the Portal Administrator role. Users who need to search for and view feeds require the PTPT1000 – PeopleSoft User permission list, which is included in the PeopleSoft User role.	See *PeopleTools 8.51 PeopleBook: Security Administration*, "Setting Up Permission Lists," Granting Access to Components and Pages. See *PeopleTools 8.51 PeopleBook: Security Administration*, "Setting Up Roles," Assigning Permissions to Roles. See *PeopleTools 8.51 PeopleBook: Security Administration*, "Administering User Profiles," Setting Roles.
Using psadmin, activate the Pub/Sub servers on the application server. This step is required only when scheduled feed messages are published to Integration Broker queues.	See *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Using PSADMIN Menus," Accessing the Application Server Options.
Configure Process Scheduler. This step is required only when scheduled feed messages are published to Integration Broker queues through an Application Engine program.	See *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Using PSADMIN Menus," Using the Process Scheduler Menu.

Using PeopleTools-Delivered Feed Data Types

PeopleTools delivers predefined feed data types for query feeds, worklist feeds, Integration Broker generic message feeds, and lists of published feeds. Follow these steps to use an existing feed data type:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Step	Reference
Publish *list of feeds* feed for each feed data type using the Define Feed Data Types page. Also, publish a list of feeds feed for the FEED data type to create a *master list* of feed lists.	See Chapter 4, "Creating and Using Feeds," Publishing a List of Feeds Feed, page 38.
Create query feeds.	You can create query feeds from Query Manager pages. See Chapter 7, "Creating and Using Query Feeds," page 65.
Create worklist feeds.	You can create worklist feeds from the Worklist and Worklist Details pages. See Chapter 8, "Creating and Using Worklist Feeds," page 83.
Create feeds for Integration Broker asynchronous, one-way service operations—also known as Integration Broker generic message feeds.	You can create Integration Broker generic message feeds from the Define IB Generic Message Feed page. See Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," page 59.

Developing New Feed Types

Use these steps to develop new feed data types:

Step	Reference
Analyze requirements for new feed data type.	See Chapter 9, "Developing New Feed Data Types," Analyzing Requirements for New Feed Data Types, page 91.
Implement feed data source application class.	See Chapter 9, "Developing New Feed Data Types," Creating the Feed Data Source Application Class, page 92.
Define feed data type.	See Chapter 9, "Developing New Feed Data Types," Defining the Feed Data Type, page 94.
Update the property maintenance page to include the Publish as Feed link.	See Chapter 9, "Developing New Feed Data Types," Adding the Publish as Feed Link to a Page, page 98.
Add standard Publish as Feed pages to a component as hidden pages.	See Chapter 9, "Developing New Feed Data Types," Adding the Four Standard Publish as Feed Pages, page 98 .
Update the view content page to include the related feeds hover menu.	See Chapter 9, "Developing New Feed Data Types," Adding the Related Feeds Hover Menu to Pages, page 104.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 2

Understanding the Feed Publishing Framework

This chapter discusses:

- Feed Publishing framework.
- Feed data types.
- Feed types and options.
- Feed security.
- Feed publication.
- My feeds.

Feed Publishing Framework

The Feed Publishing Framework provides:

- A set of APIs and code samples to assist application developers in creating new type of feeds and integrating them with existing features.
- Unified interfaces for content owners and administrative users to create, configure, and maintain feed definitions of various types.
- Interfaces for end users to discover related feeds and search feed definitions.
- Multiple language support.

The following diagram shows the Feed Publishing Framework architecture. In this diagram, use the following legend:

Orange boxes - Integration Broker components.

Light blue boxes - Portal components or GUI.

Yellow boxes - Data storage.

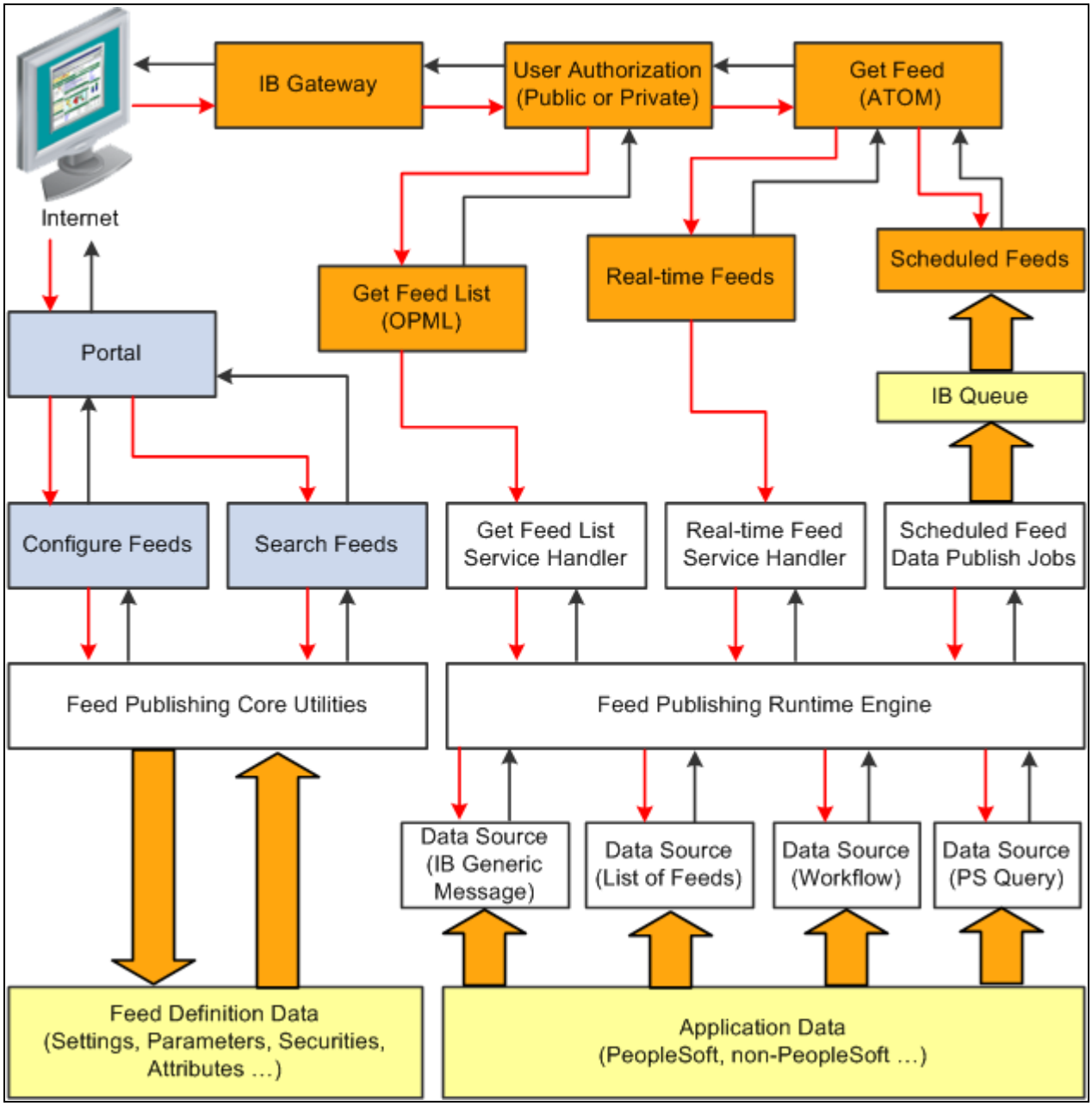
White boxes - Feed framework core.

Red lines - Incoming request.

Black lines - Response.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Gold arrows - Data to and from database.



Feed Publishing Framework architecture

Feed Publishing Runtime Engine

The centerpiece of the feed document generation is the runtime engine. The runtime engine:

- Handles the feed identification and authorization.
- Sets the data source settings and data source parameter value.
- Executes the data source object to collect data.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Transforms feed data to an Atom 1.0 feed document.

Runtime Engine Process Flow

When the runtime engine receives a request, it will first locate the feed definition based on the feed ID, and then it will determine whether the current user has access to that feed. If the user has permission, the process continues. If the user does not have permission, the process stops and the system displays an error.

Next, the runtime engine creates an instance of the feed data type application class associated with the specific feed definition, restores the Data Source Setting values and the Data Source Parameter values saved with the feed definition, and sets the runtime information, such as the requested language, and so on. At this time, the runtime engine also evaluates the runtime values of data source parameters with usage types of *Admin Specified*,*System Variable*, and *User Specified.*

Then, the runtime engine instructs the feed object to collect the data and returns the feed data in an Atom 1.0 feed document. the runtime engine does not have direct knowledge about the feed data. All business logic relevant to data collection are encapsulated in the feed data type application class PeopleCode.

By default, the currently authenticated user of the request is used for gathering the feed data. Different users requesting the same feed may receive different feed data based on their permission. This behavior can be overridden by specifying the Feed Authorization options when you create a feed definition; that is, a user ID can be specified alone with a selection that specifies whether to use this user to generate feed documents for all requests of that feed or only for anonymous requests. This option is a per feed definition, and it will be stored with the feed definition.

Feed Document Generation and Delivery

By default, all feeds managed by the framework will be served as real-time feeds through Integration Broker framework by using the GetFeed service operation.

GetFeed Service Operation

A GetFeed service operation (PTFP_GETFEED) uses the runtime engine to locate the feed definition and generate feed document in Atom 1.0 format. Transformation can be done in the service operation level to get feed documents in other formats. Integration Broker manages user authentication, caching, and feed data type service operation-level security.

The primary task of the GetFeed service operation handler is to collect the request information and then forward them to the runtime engine. The handler also catches the exceptions raised by the runtime engine, and reports them in the specific way applicable to Integration Broker.

By default, the GetFeed service operation only has one required parameter, the feed ID, and three optional parameters, the language code, the portal name, and the node name. The feed URL given out by the framework contains these parameters only, when applicable.

When the feed request comes in, the Integration Broker copies the values of the query parameters to the corresponding fields defined in the default feed request message definition. Then, the service operation passes this information to the runtime engine for further processing. If other query parameters exist in the request URL, the service operation collects and passes them to the runtime engine. These additional query parameter values override the values of Data Source Parameters for which their usage types are *User Specified.*

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Any feed data type can override this default behavior by providing its own feed request message definition extended from the default one, as well as by using a different method for generating the feed URL to include more default query parameters. Regardless, the feed ID query parameter should always be a required parameter. We do not recommend the use of query parameters for data source settings or non-user-specified data source parameters.

Feed Document Properties

The runtime engine will return an Atom 1.0 feed document. The properties of the feed definition become the feed header properties of the feed document. Each item in the feed-format-neutral object collection becomes one entry of the feed document. The data mapping relationships are listed below.

Feed Element Mapping

The feed data source uses a feed-format-neutral object collection to collect data. This table describes the mapping between the Atom 1.0 feed-level elements and the feed definition properties:

Atom 1.0 Feed-Level Element	***Feed Definition Property***
author	Author
category	Feed data type, and feed category
contributor	Contributor
generator	Latest entry in PSRELEASE table
icon	Icon
id	System generated URL to open the feed
link	Alternate link: The content web page URL associated with the feed, which is provided by the data source. Self link: System generated URL to open the feed
logo	Logo
rights	Copyright
subtitle	Feed description
title	Feed title
update	Current date and time of request

This table describes the mapping between the Atom 1.0 entry-level elements and properties of items in the feed-format-neutral object collection. The real values of these properties are determined by the data source at runtime.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Atom 1.0 Entry-Level Element	*Item Property*
author	Author
category	Category **Note.** This item allows multiple entries.
content	FullContent, if available
contributor	Contributor **Note.** This item allows multiple entries.
id	GUID, if available or content URL
link	Alternate link: contentURL Other links: enclosure, if available. **Note.** This item allows multiple entries.
published	Date and time published
rights	Copyright
source	N/A
summary	Description
title	Title
updated	Date and time updated

Note. If any property contains empty values, the corresponding element is not added.

Feed Data Type Application Classes

The feed data type application class bridges the data and the feed definition. It has two roles:

- At design time, it provides information to the framework about how to define the feed definition for this type of data.
- At runtime, the framework uses the application class to collect feed data.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Design Time Role of Application Classes

At design time, the feed data type application class provides information of a specific type of data to the framework, which includes list of data source settings, list of data source parameters and their default values, data security, and name of the default feed service definition. The feed data type application class handles events such as processes that occur when you delete a feed definition. The framework uses all of this information in the feed definition creation and maintenance.

Every feed data type may have zero to three data source settings. The data source settings are used to uniquely define the feed data source of the given type of data. You must define and store the values of the data source settings with the feed definition. The feed data type application class can also provide the prompt information for each data source setting.

Every feed data type may have zero or more data source parameters. The data source parameters are used to fine tune the feed or personalize the feed, for example, a Company News feeds should include all child sections or a DEMO workspace feed should not include discussion data. Values of data source parameters will be determined at runtime based on its usage type—fixed value, system variable value, user specified value, and so on. The default values of data source parameters are stored with the feed definition. The Feed Data Type application class could provide the description, prompt information, default values, and default usage type for each data source parameter.

Run Time Role of Application Classes

At runtime, the runtime engine finds the feed definition and the associated feed data type application class based on the requested feed ID. It creates an instance of the feed data type application class associated with the specific feed definition, restores the data source setting values and the data source parameter values saved with the feed definition, evaluates the runtime values of those data source parameters based on their usage type, and sets other runtime information. It then executes the object to retrieve the content data of the feed.

The feed data type application class uses a feed-format-neutral object collection to temporarily store the data. It does not transform data to feed document directly. This design enables the application developers to expand the object model or use their own data objects. The advantages using a feed-format-neutral object collection instead of the feed-format-specific XML document are:

- It shields you from having to deal with complex details of specific feed format.
- It minimizes the possibility of using a wrong XML element or structure.
- It standardizes the use of specific elements within the feed.
- It enables easier migration to newer feed format standards or a completely different feed format.

Creating a New Feed Data Type

To create a new type of feeds, application developers implement a new Feed Data Type application class from the base class provided by the framework, and they associate it with one feed data type service definition. This application class encapsulates all business logics about how to define the feed definition and how to gather feed data. At runtime, it uses a feed-format-neutral object collection to temporary store the data. It does not transform data to feed document directly.

Additional development work will be required to enable Content Owners to publish contents as feeds directly from content maintenance pages and to add feed subscription links to view content pages.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

See Also

Chapter 9, "Developing New Feed Data Types," page 91

Feed Data Types

This section discusses the four delivered feed data types:

- List of feeds (FEED).
- Integration Broker generic message feeds (GENERICFEED).
- PeopleSoft Query feeds (PSQUERY).
- SES feed data source feeds (PTSF_SES_FEED_DT).
- Worklist feeds (WORKLIST).

List of Feeds (FEED)

A *list of feeds* feed enables feed administrators to generate a feed that displays a list of all feeds of a specific feed data type.

See Chapter 4, "Creating and Using Feeds," Publishing a List of Feeds Feed, page 38.

Integration Broker Generic Message Feeds (GENERICFEED)

Integration Broker generic message feeds enable administrators to expose Integration Broker messages used in asynchronous, one-way service operations as feeds.

See Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," page 59.

PeopleSoft Query Feeds (PSQUERY)

PeopleSoft Query feeds enable query administrators to expose query outputs as feeds.

Note. Any user with access to Query Manager can publish query feeds.

See Chapter 7, "Creating and Using Query Feeds," page 65.

Worklist Feeds (WORKLIST)

Worklist feeds enable workflow administrators to expose worklists as feeds.

See Chapter 8, "Creating and Using Worklist Feeds," page 83.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

SES Data Source Feeds (PTSF_SES_FEED_DT)

SES feeds are used internally by PeopleTools as a search data source.

Feed Types and Options

This section discusses:

- Real-time feeds.
- Scheduled feeds.
- Paged feeds.
- Incremental feeds.

Real-Time Feeds

Real-time feeds are dynamic—that is, produced when the user requests them. Real-time feeds are created by using Integration Broker synchronous service operations. These service operations are similar to other Integration Broker service operations except that the service operation handler returns an ATOM_1_0 message.

Whenever the HTTP listening connector gets a request for a real-time feed, it invokes the appropriate synchronous service operation. It uses either the PS_TOKEN or basic authentication credentials. User authentication and service operation authorization are handled by Integration Broker; feed authorization is handled by the Feed Publishing Framework. If the user has access to the feed, the service operation handler adjudicates any HTTP request parameters passed to it and generates an Atom feed and returns it in an ATOM_1_0 response message.

Scheduled Feeds

Scheduled feeds are published asynchronously and stored as messages in Integration Broker queues.

Scheduled feeds can be further classified into up-front feeds and generic feeds. When using up-front feeds, the messages published to the Integration Broker queues are feed messages. When using generic feeds, the messages published to Integration Broker queues are either PeopleSoft rowset or XML messages. They are not feed (Atom) XML messages.

When a user requests the feed, the GetFeed (PTFP_GETPREPUBFEED) synchronous service operation is invoked by the HTTP listening connector. The GetFeed service operation handler fetches the appropriate feed messages from Integration Broker queues and collates them into a single feed message. The GetFeed service operation handler collates the messages for up-front feeds into a single feed message while, for the generic feeds, it wraps the feed element tags to the Integration Broker messages and then collates them into a single feed message.

Note. Unlike real-time feeds, for which you can create your own service operation to deliver the feed, scheduled feeds always use the PTFP_GETPREPUBFEED service operation to deliver feeds.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Paged Feeds

A *paged feed* is a feed that has been split into pages (also known as segments) to improve system performance in delivering large feed documents and to improve performance for consuming a feed. A paged feed is presented with first, last, next, and previous links to allow access to additional pages in the feed document.

Paged feeds are supported for scheduled feeds only. The framework supports paged feeds via Integration Broker message segments. %MaxMessageSize is recommended when creating Integration Broker message segments for paged feeds.

When setting a feed's paging property, one can select either *No Paging* or *Segmented*, which determine how the framework displays the complete feed:

- Segmented
  - This option is designed for feeds intended for crawlers or system synchronization.
  - This option is not supported by most feed readers or clients.
  - Feed links (first, next, previous, last) are added to the feed XML.
  - Feed entries are not restricted by the Max Rows Limit parameter.

    See Chapter 5, "Administering Feeds," Setting Feed Publishing Framework Options, page 48.
- No paging
  - This option is designed for feeds intended for end user viewing.
  - This option is supported by all feed readers or clients.
  - Feed entries are restricted by the Max Rows Limit parameter.

The following table illustrates how paged feed options and Integration Broker message segment options affect the output of the framework:

Integration Broker Message Option	***Feed Option – Segmented***	***Feed Option – No Paging***
Segmented	• Each message segment becomes a page in the feed XML. • Oldest message contents are displayed first. • The Max Rows Limit is not applicable.	• All message or message segments are displayed in a single feed XML. • Latest message contents are displayed first. • The Max Rows Limit applies.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Integration Broker Message Option	*Feed Option – Segmented*	*Feed Option – No Paging*
Non-segmented	• Each message becomes a page in the feed XML. • Oldest message contents are displayed first. • The Max Rows Limit is not applicable.	• All message content entries are displayed in a single feed XML. • Latest message contents are displayed first. • The Max Rows Limit applies.

See Also

Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," Publishing Integration Broker Generic Message Feeds, page 60

PeopleTools 8.51 PeopleBook: PeopleCode API Reference, "Feed Classes," DSPARAMETER_SF_PAGING

PeopleTools 8.51 PeopleBook: PeopleCode API Reference, "Feed Classes," SF_PAGINGOPTION_NOPAGING

PeopleTools 8.51 PeopleBook: PeopleCode API Reference, "Feed Classes," SF_PAGINGOPTION_SEGMENTED

Incremental Feeds

An *incremental feed* is a feed that has been published and updated with time stamps that allow the feed content to be delivered incrementally. An incremental feed allows the Feed Publishing Framework to deliver only the feed content that has changed since the feed was last requested by the user.

Note. The incremental feed option is incompatible with the paged feed option. For the delivered feed data types that can be specified as paged, the *Incremental* option is disabled when *Segmented* is selected.

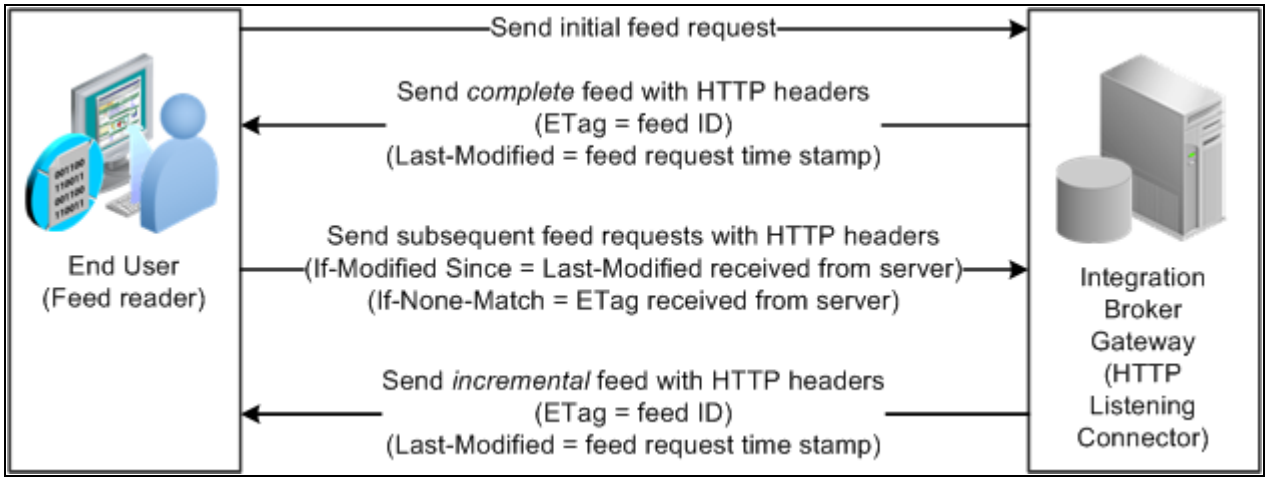
Incremental feeds save network bandwidth by using HTTP conditional GET headers. The HTTP conditional GET headers supported are:

- ETag
- If-None-Match
- Last-Modified
- If-Modified-Since

Note. Not all feed readers support incremental feeds, which requires that the reader retain the feed request timestamp and present that data as part of the next feed request.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

The following diagram illustrates how HTTP conditional headers are used with incremental feeds. With the initial feed request, the feed reader does not include any HTTP conditional headers. In the response, the PeopleSoft system sends the *complete* feed data and includes two HTTP conditional headers: ETag equals the feed ID and Last-Modified equals the feed request time stamp. When the feed reader makes a subsequent feed request, it includes two HTTP conditional headers: If-None-Match equals the ETag sent by the PeopleSoft system and If-Modified-Since equals Last-Modified sent by the system. In the response, the PeopleSoft system sends just the *incremental* feed data and includes the same two HTTP conditional headers as the initial response; Last-Modified now represents the latest feed request time stamp.



Incremental feeds

The Feed Publishing Framework provides out-of-the-box support for scheduled, incremental feeds. For the feed data source, you must define the DSPARAMETER_INCREMENTAL data source parameter and assign an appropriate value to it. For real-time, incremental feeds, you must also define the DSPARAMETER_INCREMENTAL data source parameter. In addition, you must implement the PeopleCode to deliver an incremental feed in the data source's execute method.

See Also

Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," Publishing Integration Broker Generic Message Feeds, page 60

PeopleTools 8.51 PeopleBook: PeopleCode API Reference, "Feed Classes," DSPARAMETER_INCREMENTAL

PeopleTools 8.51 PeopleBook: PeopleCode API Reference, "Feed Classes," INCREMENTALOPTION_NO

PeopleTools 8.51 PeopleBook: PeopleCode API Reference, "Feed Classes," INCREMENTALOPTION_YES

Feed Security

This section discusses security for:

- Creating feed data types.
- Publishing feeds.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Viewing feeds.

Security for Creating Feed Data Types

Security for creating new feed data types is based on permission lists. To create new feed data types, the user must be authorized to access pages in the PTFP_DATATYPE component on the PTFP_FEED_PUBLISHING menu.

Note. Users with access to the PTPT1300 (Portal Administrators) permission list automatically have access to these pages.

See *PeopleTools 8.51 PeopleBook: Security Administration*, "Setting Up Permission Lists."

Security for Publishing Feeds

You publish feeds by accessing the hidden Publish as Feed pages when you click the Publish as Feed link on a page. This link is located on different pages in the applications based on the type of feed. The link is currently available through:

- Workflow pages (WORKLIST and WORKLIST_DETAILS).
- Query Manager pages (QUERY_MANAGER).
- Define IB generic message feed pages (PTFP_GENERIC_FEED).
- Define feed data type pages (PTFP_DATATYPE).

To access the link, you must be authorized to access the specific application pages where you find the link.

Security for Viewing Feeds

The two levels of feed security are:

- Feed-level security.
- Data-level security.

Feed-Level Security

The feed level security is managed by the Feed Publishing Framework. Feed level security determines which feeds are visible to the user when accessing the My Feeds page or any related hover menus. You can configure feed security to be:

- Public.

  Public feeds run under context of the default user that is associated with the ANONYMOUS node.

  See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Adding and Configuring Nodes," Defining Node Parameters.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Realtime.

  Every time a user accesses a feed during search or execution, the data source object determines whether the current user has access to the feed. This security option has an advantage in that the feed security is always in sync with the data source. This security option can greatly impact performance of feed searches and should only be used when the data security is constantly changing, or the data security could not be defined using role or permission list based security model.

- Permission list and roles.

  You assign access to the feed based on permission lists and roles.

Data-Level Security

The data level security is checked by each data type supporting application class when the runtime engine executes it to collect feed data. It is always checked in real time. Users who have access to a feed but not the data will receive a feed document that contains no entries. Different users who subscribe to the same feed might receive different feed data depending on the permissions that they have. It is possible to sync the feed data security to the feed definition using the Publish Feed Definition pages.

Important! The developer is responsible for building data level security into the data source application class logic; data level security is not automatic.

Feed Publication

You use the Publish as Feed link to publish data as a feed. The Publish as Feed link provides access to the four feed publishing pages:

- Publish Feed Definition (PTFP_PUB_AS_FEED).
- Advanced Feed Options (PTFP_PUB_AS_ADVOPT).
- Publish as Feed (PTFP_PUB_AS_LIST).
- Publish Feed Definition to Sites (PTFP_PUB_AS_SITES).

Note. The framework provides these pages; however, each data type might alter or replace them as necessary.

See Chapter 4, "Creating and Using Feeds," Publishing Feeds, page 31.

My Feeds

End users can search and view feeds by using the My Feeds page (PTFP_VIEW) link, which you find in the menu navigation. Search for feeds specific to the user and then click the feed document link to view it in a new browser window. You can also add the feed URL to feed readers; or you can export the search results to an OPML (Outline Processor Markup Language) file and save the list for later use.

See Chapter 4, "Creating and Using Feeds," Using the My Feeds Page, page 39.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 3

Configuring Your PeopleSoft System to Support Feeds

To use the Feed Publishing Framework, you must first configure your PeopleSoft system to support feed publication and consumption.

This chapter discusses how to:

- Configure the Integration Broker gateway and target nodes.
- Configure Integration Broker service target locations.
- Configure the default user of the ANONYMOUS node.
- Configure the default local node.
- Set URI text for local host nodes.

See Also

PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker, "Integration Scenarios," Understanding Integration Setup

Configuring the Integration Broker Gateway and Target Nodes

To configure the Integration Broker gateway and target nodes:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

1. Select PeopleTools, Integration Broker, Configuration, Quick Configuration.

   The Integration Broker Quick Configuration page is displayed.

   See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Using the Integration Broker Quick Configuration Page."

   a. Enter the machine name and complete URL to the PeopleSoftListeningConnector in the Gateway URL field:

   Integration Broker Quick Configuration page, Local Gateway group box

   b. Click the Ping Gateway button.

      The status should return as active.

   c. In the Integration Broker Domains group box, set the status for this machine to *Active*.

   d. Also in the Integration Broker Domains group box, set the status for any pub/sub domains to Active to support scheduled feeds.

      Note. The pub/sub domain must be enabled first in the application server configuration through psadmin.

      ---

   e. Click the Save button to save your changes.

2. Click the Advanced Gateway Setup link.

   The Gateways page is displayed.

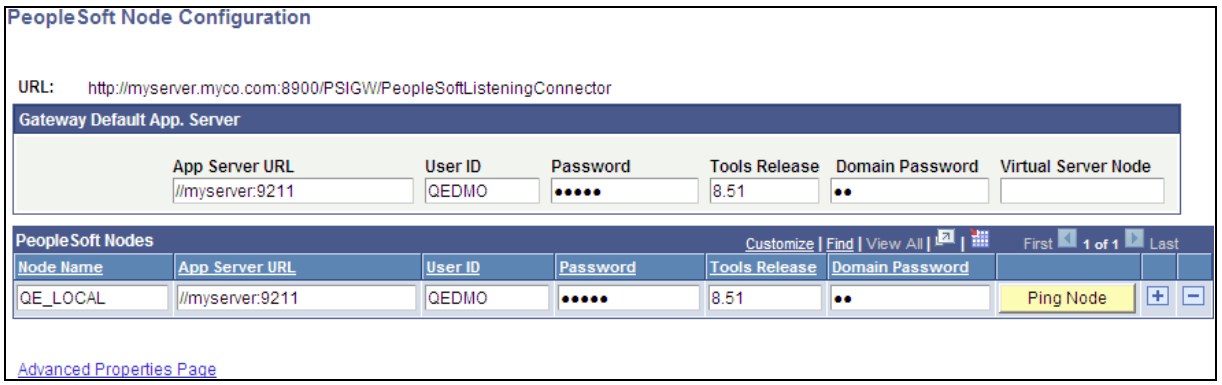
Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

3. Click the Gateway Setup Properties link and then login.

   The PeopleSoft Node Configuration page is displayed.

   See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Setting Oracle Jolt Connection Properties.

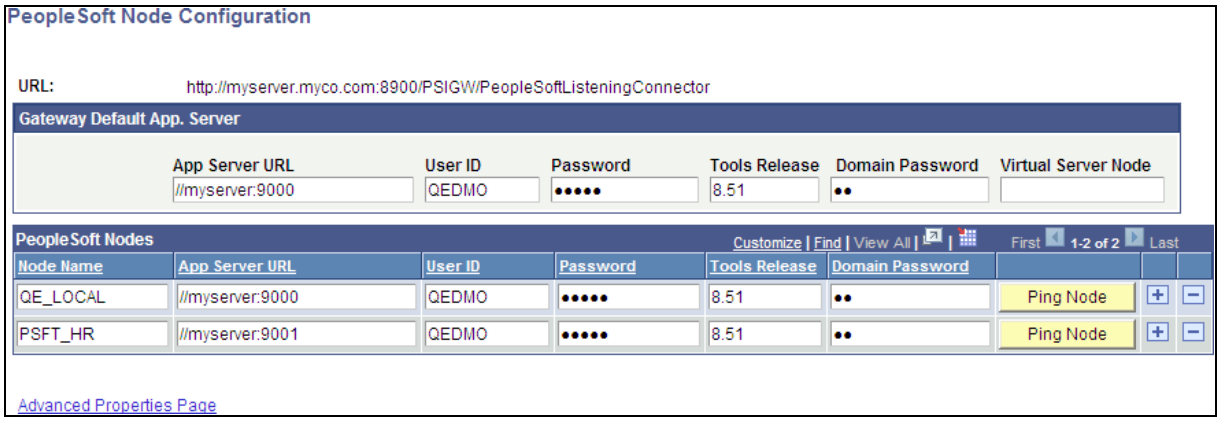
   a. Enter values for your environment. Enter a default target node in the Gateway Default App Server group box and the local target node in the PeopleSoft Nodes group box.



   PeopleSoft Node Configuration page

   b. For a shared gateway, also enter remote target nodes (the default local node on the remote system) in the PeopleSoft Nodes group box.

   The following example shows a shared gateway configuration:



   PeopleSoft Node Configuration page (shared gateway configuration)

   c. Click the Save button.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

d. Click the Ping Node button for each node.

The status should return as success.

Note. If you receive an Integration Broker authentication error, the default local node requires that the authentication option be set to password or certificate. See the troubleshooting appendix for more information.

See Appendix A, "Troubleshooting Tips," page 115.

e. On the Ping Node Results page, click the Return button.

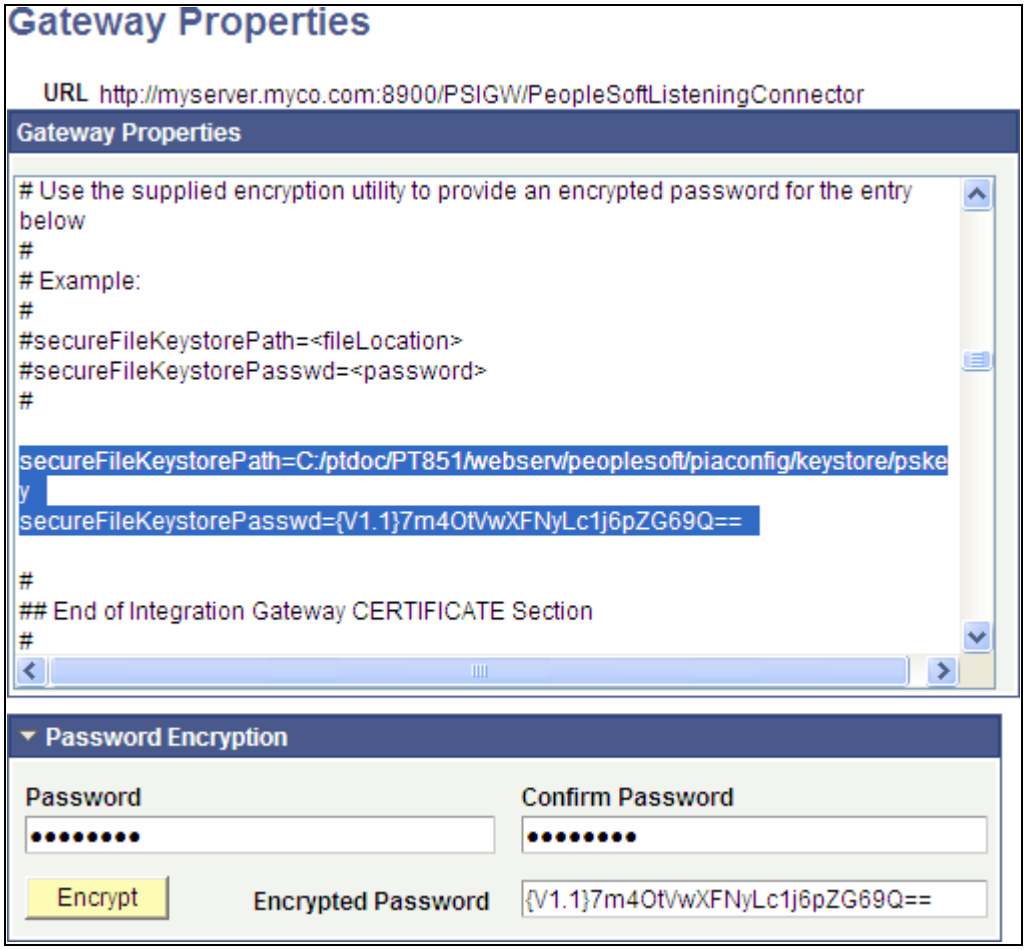
 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

4. On the PeopleSoft Node Configuration page, click the Advanced Properties Page link.

   The Gateway Properties page is displayed.

   See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Using the integrationGateway.properties File.

   a. Enter the full path to the keystore file and the encrypted keystore password:



   Gateway Properties page (setting the keystore location and encrypted password)

   See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Configuring Security and General Properties and *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Encrypting Passwords.

   b. Click the OK button.

5. On the PeopleSoft Node Configuration page, click the Save button again.

Configuring Integration Broker Service Target Locations

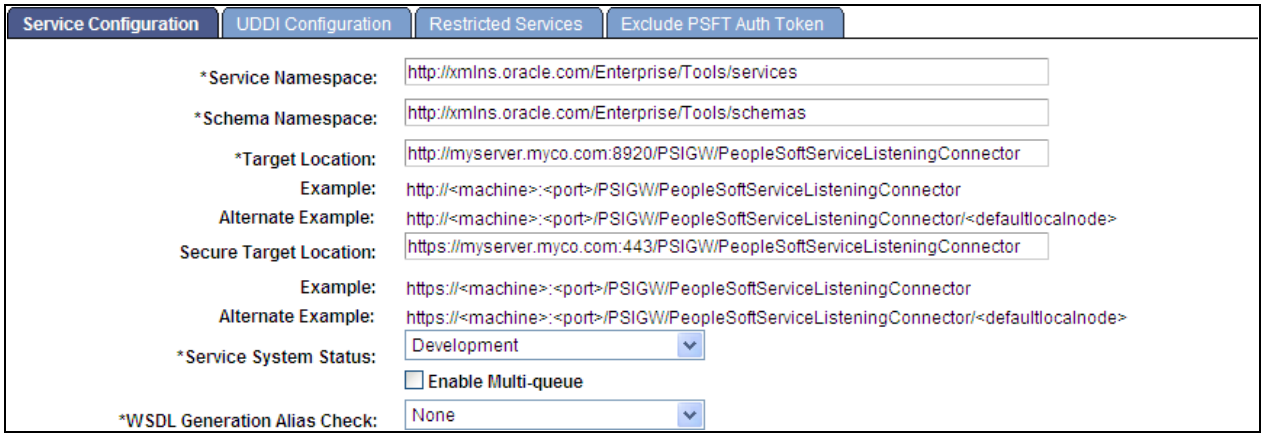
To configure service target locations:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

1. Select PeopleTools, Integration Broker, Configuration, Service Configuration.

   See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Configuring PeopleSoft Integration Broker for Handling Services," Setting Service Configuration Properties.

2. Enter a value for the target location (or locations) pointing to the appropriate Integration Broker gateway connector:



Service Configuration page

   - If you plan to use basic authentication only and SSL has not been configured on your system, then enter a value in the Target Location field only.
   - If you plan to use secure authentication and SSL has been configured on your system, then enter a value in the Secure Target Location field.

   Important! Oracle recommends that you use the secure authentication approach.

   Also note, if you specify both target locations, then the secure target location is always used for access to feeds.

   More information on setting up SSL is available in PeopleBooks.

   > See *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Working with Oracle WebLogic," Implementing WebLogic SSL Keys and Certificates and *PeopleTools 8.51 PeopleBook: System and Server Administration*, "Working with IBM WebSphere," Setting Up SSL For WebSphere.

3. Click the Save button to save the configuration.

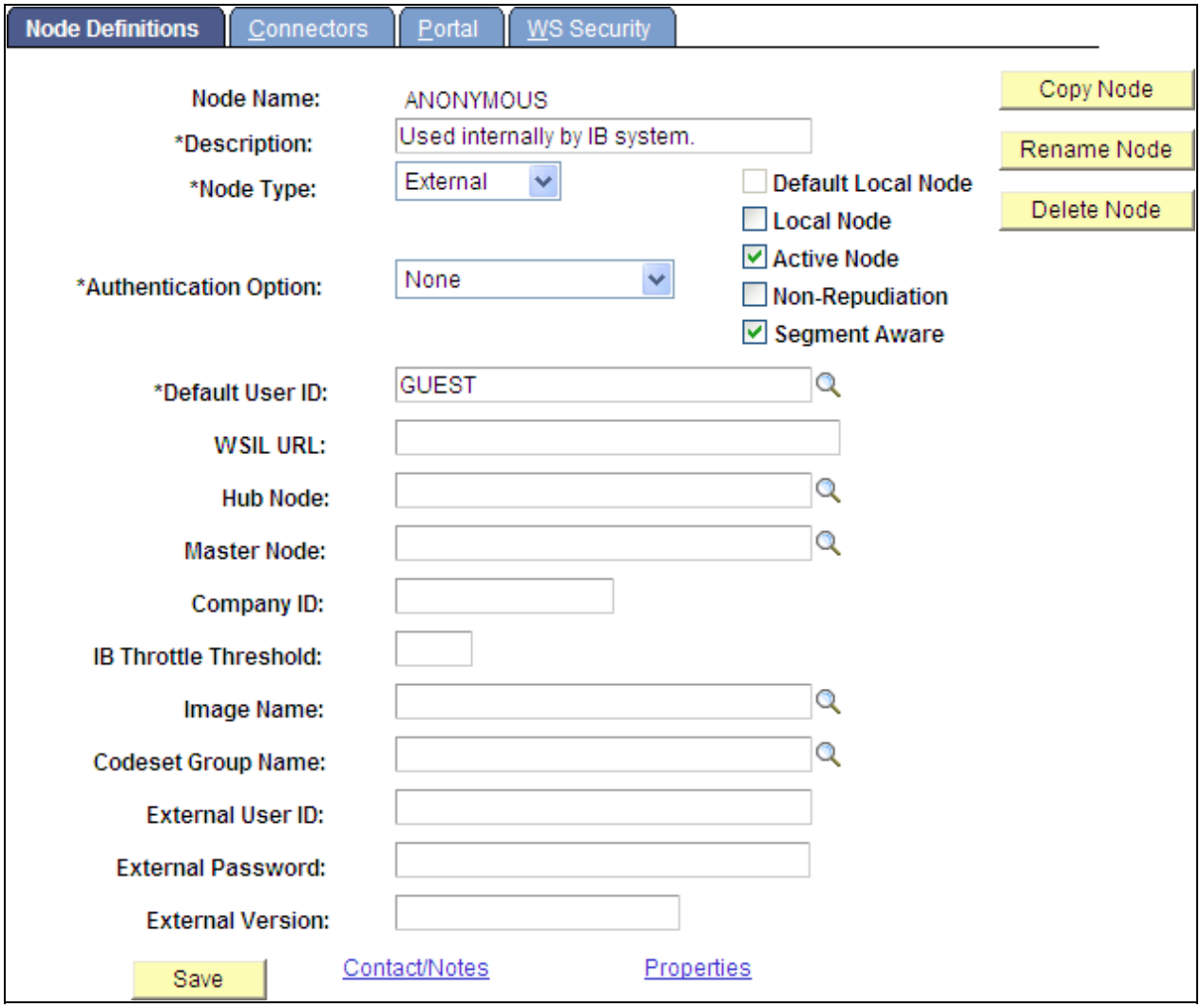
Configuring the Default User of the ANONYMOUS Node

To configure the default user of the ANONYMOUS node:

1. Select PeopleTools, Integration Broker, Integration Setup, Nodes.
2. Select the ANONYMOUS node.
3. Select the Node Definitions page.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

4. Verify that the Default User ID field contains a valid user with limited privileges—for example, the GUEST user:



Node Definitions page – ANONYMOUS node

See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Adding and Configuring Nodes," Defining Node Parameters.

Configuring the Default Local Node

To configure the default local node:

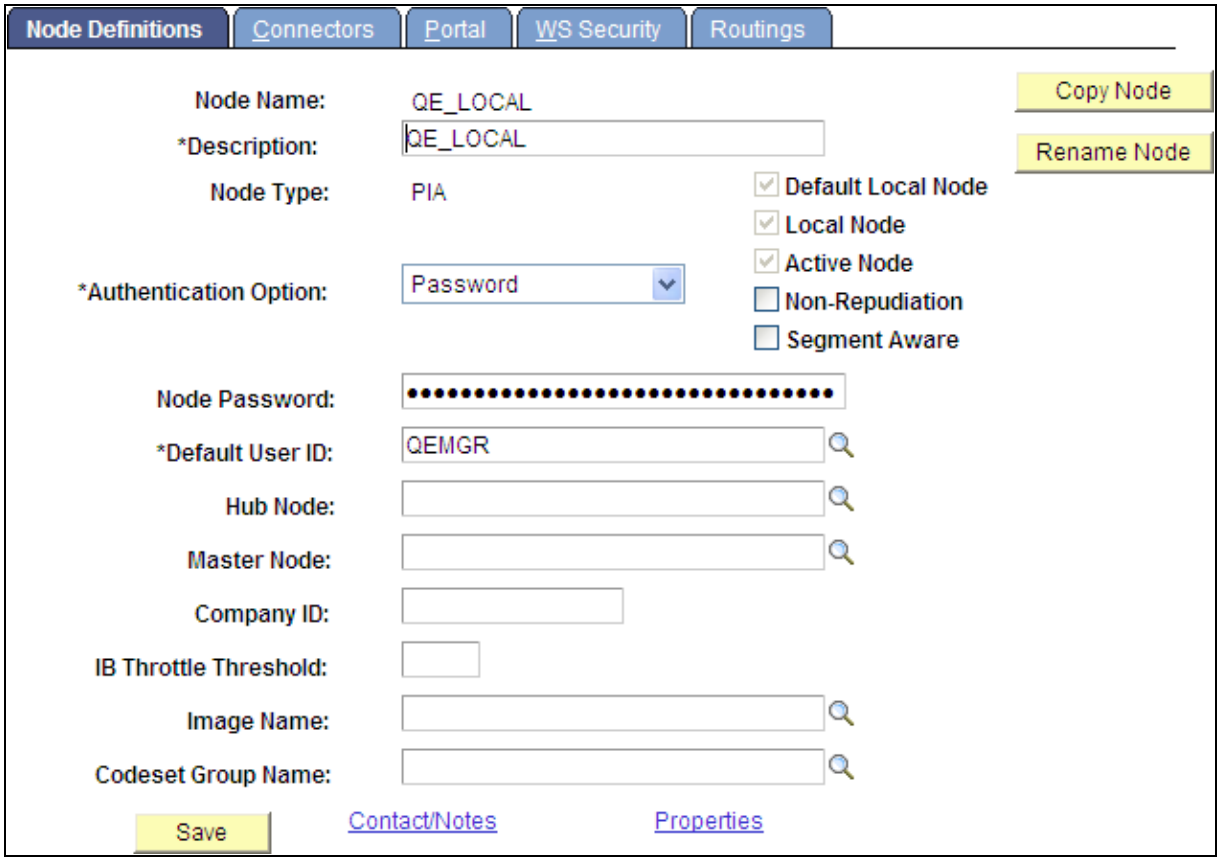
1. Select PeopleTools, Integration Broker, Integration Setup, Nodes.
2. Select the default local node.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

3. Select the Node Definitions page.

   a. Verify that the node type is PIA and that the Authentication Option field value is *Password* or *Certificate*.

   b. Verify that the node password and default user ID are set:



Node Definitions page – Default local node

See *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Setting Up Secure Integration Environments," Implementing Node Authentication and *PeopleTools 8.51 PeopleBook: PeopleSoft Integration Broker Administration*, "Adding and Configuring Nodes," Defining Node Parameters.

4. Select the Connectors page:

   a. Click the Ping Node button.

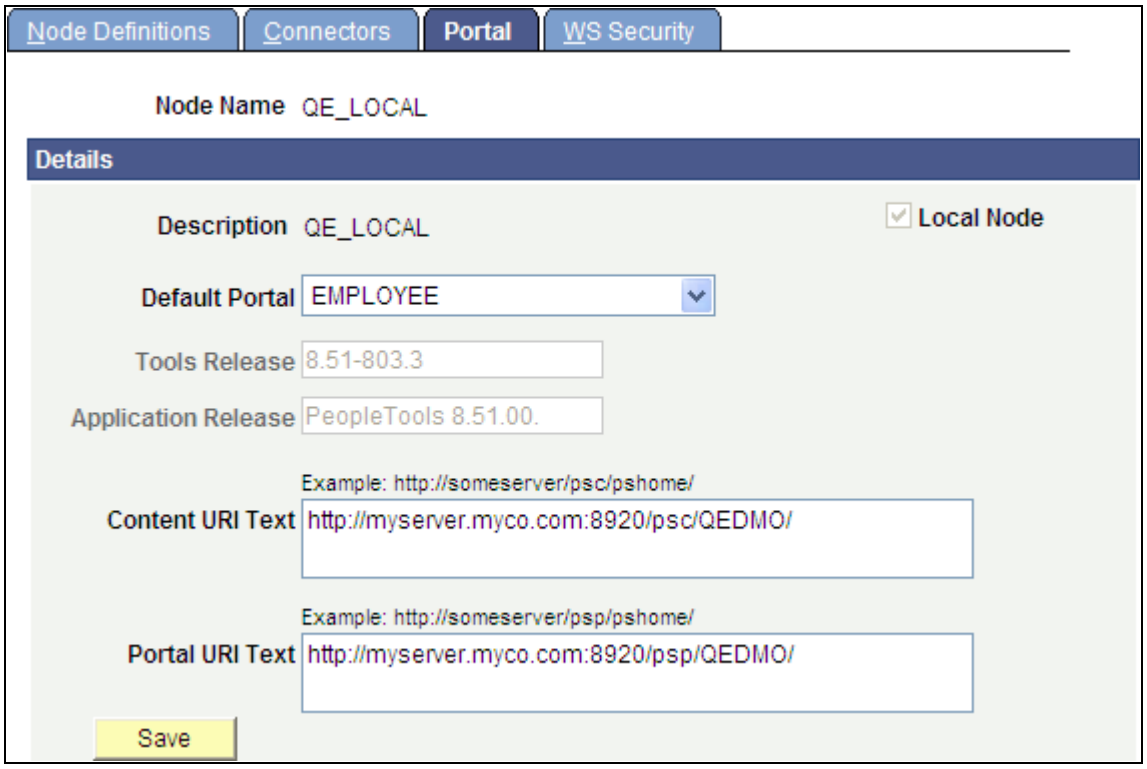
      It should show success.

   b. Click the Return button.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

5. Select the Portal page:

   a. Enter values for the Content URI Text field and the Portal URI Text field:



   Portal page – Default local node

   See *PeopleTools 8.51 PeopleBook: PeopleTools Portal Technologies*, "Configuring the Portal Environment," Setting Portal Nodes.

   b. Save the configuration of the default local node.

Important! If you set or changed the authentication option on the Node Definitions page, log out and log back in again. Otherwise, you might get an authentication token error when you attempt to access a feed.

Setting URI Text for Local Host Nodes

For each local host node, you must set the URI text for the node definition.

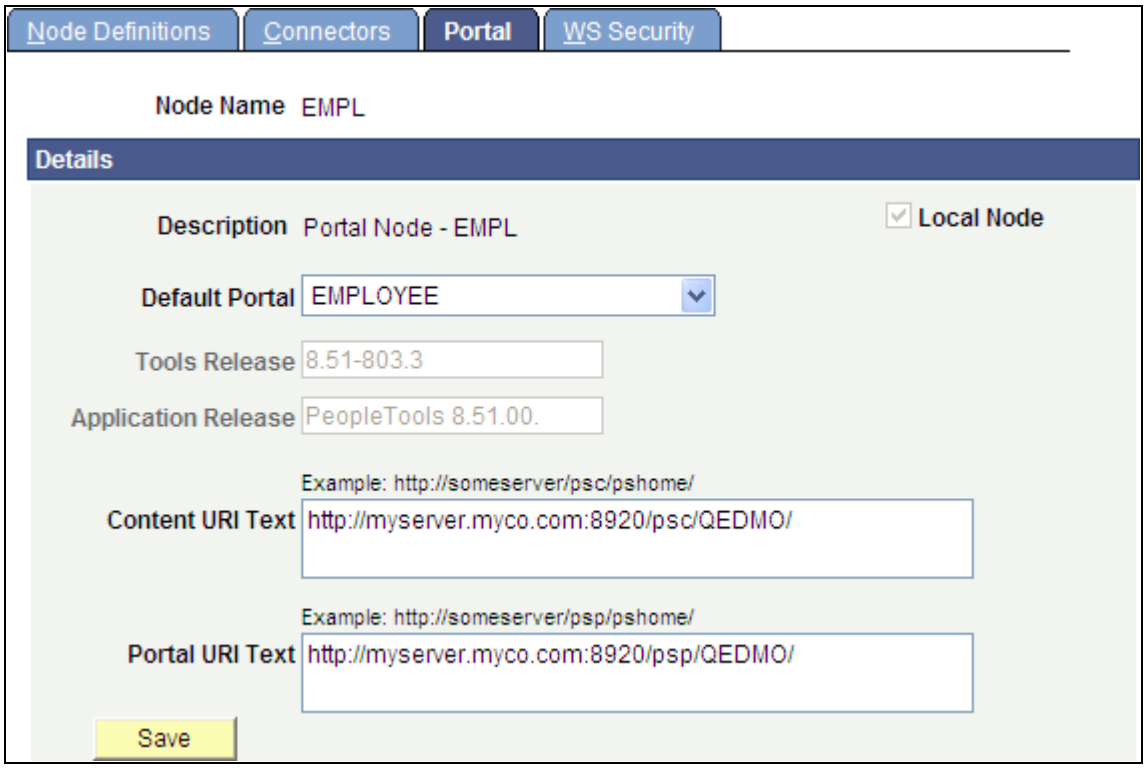
Repeat the following procedure for each local host node:

1. Select PeopleTools, Integration Broker, Integration Setup, Nodes.
2. Select a local host node.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

3. Select the Portal page:

   a. Enter values for the Content URI Text field and the Portal URI Text field:



   Portal page – Local host node

   See *PeopleTools 8.51 PeopleBook: PeopleTools Portal Technologies*, "Configuring the Portal Environment," Setting Portal Nodes.

   b. Save the configuration of the local host node.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 4

Creating and Using Feeds

This chapter discusses how to:

- Define and publish feeds.
- Publish feeds list feeds.
- Access feeds.
- View feeds

Defining and Publishing Feeds

This section presents the process flow for publishing and consuming feeds and discusses how to:

- Publish feeds.
- Define feed properties.
- Define advanced feed options.
- Manage published feeds.
- Publish feeds to additional sites.

Pages Used to Publish Feeds

Page Name	*Definition Name*	*Navigation*	*Usage*
Publish Feed Definition	PTFP_PUB_AS_FEED	• Click the Publish as Feed link on the administration page for the item to be published as a feed. • Click the Edit button on the Publish as Feed page. • Click the Add Feed button on the Publish as Feed page.	Define feed security options, enter optional feed properties, and access advanced options.

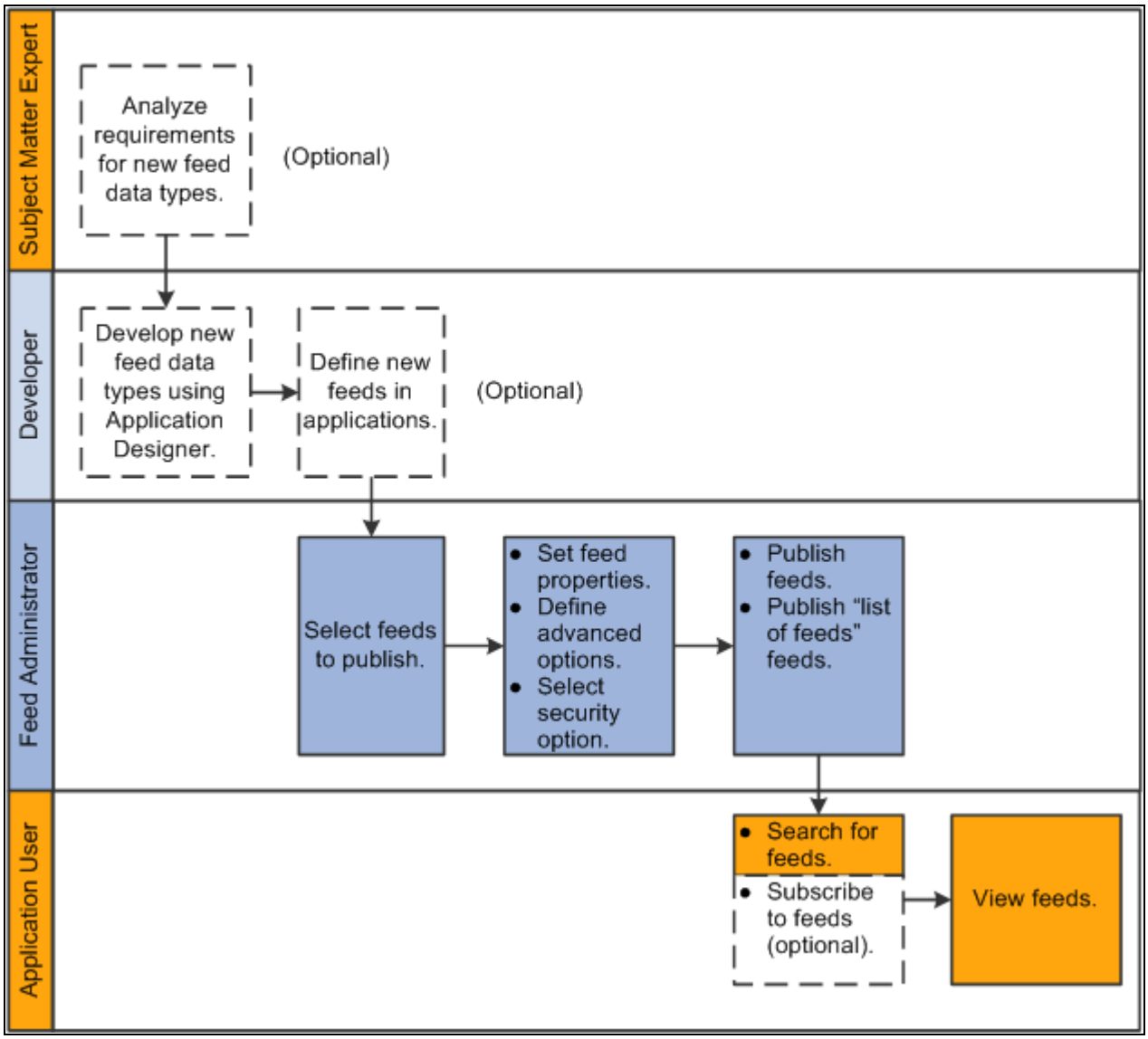
Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Page Name	*Definition Name*	*Navigation*	*Usage*
Advanced Feed Options	PTFP_PUB_AS_ADVOPT	Click the Advanced Options link on the Publish Feed Definition page.	Enter advanced option values that are specific to the feed data type.
Publish as Feed	PTFP_PUB_AS_LIST	• When one or more feeds have been defined for this item, click the Publish as Feed link on the administration page for the item. • Click the Publish button on the Publish Feed Definition page. • Click the Cancel button on the Publish Feed Definition page.	Review, edit, add, or delete feed definitions for this item.
Publish Feed Definition to Sites	PTFP_PUB_AS_SITES	Click the Publish Feed to Other Sites link on the Publish Feed Definition page.	Select additional sites to publish an existing feed definition.

Process Flow for Feed Publishing and Consumption

This diagram illustrates the business process flow for developing, administering, and viewing feeds.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Business process flow for developing, administering, and viewing feeds

Publishing Feeds

To publish a feed, you access "Publish as Feed" pages delivered in PeopleTools. In a PeopleSoft system, administrative users and content owners can use the Publish as Feed pages to create, view, edit, and delete feed definitions related to a particular content item. These pages are accessed from the content maintenance pages for that type of content.

Each feed data type has its own set of properties necessary to define feed details. This table lists the base feed data types delivered in PeopleTools, and the navigation path to the publish pages for each data type.

Feed Data Type	***Navigation***
FEED	PeopleTools, Feeds, Define Feed Data Types See Chapter 4, "Creating and Using Feeds," Publishing a List of Feeds Feed, page 38.

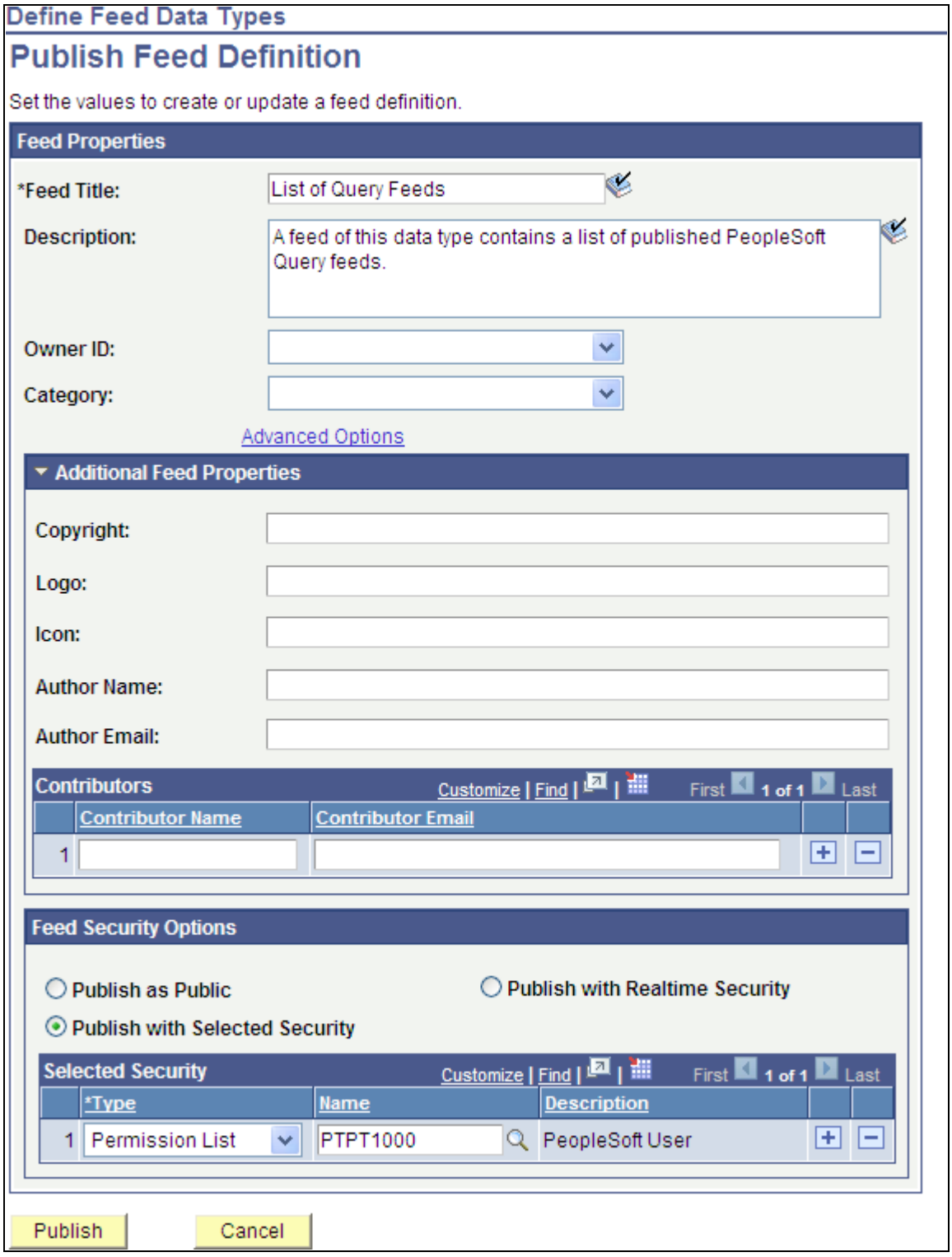
Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Data Type	*Navigation*
GENERICFEED	PeopleTools, Feeds, Define IB Generic Message Feed See Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," page 59.
PSQUERY	Reporting Tools, Query, Query Manager See Chapter 7, "Creating and Using Query Feeds," page 65.
WORKLIST	Worklist, Worklist Worklist, Worklist Details See Chapter 8, "Creating and Using Worklist Feeds," page 83.

Defining Feed Properties

Access the Publish Feed Definition page (click the Publish as Feed link).

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Publish Feed Definition page (for a list of feeds type feed)

When you click the Publish as Feed link, you then must define feed properties such as the feed title, security, and other options.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Properties

Feed Title This field is pre-populated based on the selected content item information when you create a new feed definition. You can edit the field. Enter a maximum of 30 characters.

Description Enter no more than a 254-character description of the feed.

Owner ID Select the owner ID for the feed definition.

The owner ID is a way to identify which definitions are owned by which PeopleSoft applications, such as PeopleSoft General Ledger, Accounts Receivables, and so on. The values in the drop-down list box are Translate table values associated with the OBJECTOWNERID field.

Note. The owner ID does not appear in the published feed document.

Category Select a category for the feed definition from the list of active categories.

If a feed definition is assigned to a category and then that category is deleted, the feed definition no longer displays an assigned category.

Advanced Options Click the Advanced Options link to access the Advanced Options page for the feed data type. Advanced options vary by feed data type.

Additional Feed Properties

Not all feed readers display all feed properties. This list describes the additional feed properties that some feed readers process and display.

Note. Values entered in these fields replace the default feed property values defined at the feed data type level.

See Chapter 2, "Understanding the Feed Publishing Framework," Feed Document Properties, page 8.

See Chapter 9, "Developing New Feed Data Types," Defining the Feed Data Type, page 94.

Copyright Enter a copyright date to be included in the feed document.

Logo Enter a URL to the logo to be included in the feed document—for example, *http://myserver.mycompany.com:80/images/logo.gif*.

Icon Enter a URL to an icon to be included in the feed document—for example, *http://myserver.mycompany.com:80/images/icon.gif*.

Author Name Enter an author to be included in the feed document.

Author Email Enter an author email address to be included in the feed document.

Contributor Name Enter a contributor to be included in the feed document.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Contributor Email Enter a contributor email address to be included in the feed document.

Feed Security Options

This list describes the feed security options:

Note. Query feeds apply security on the Advanced Options page.

Worklist feeds support real-time security only and feed security options do not appear for Worklist feeds.

See Chapter 8, "Creating and Using Worklist Feeds," Understanding Worklist Feeds, page 83.

See Chapter 7, "Creating and Using Query Feeds," Defining Advanced Options for Query Feeds, page 67.

Publish as Public Select to make the feed available for public access.

Publish with Selected Security Select to make the feed available based on the viewer's role-based permission lists.

When you select this type of security, the Selected Security grid appears, enabling you to set role and permission list security.

Note. User-based permission lists, such as the Primary Permission List, are not applied with this type of security.

Publish with Realtime Security Select to have the system check the data source object to determine whether the viewer has access to the feed in real time. This option is the default for new feed definitions.

Sync with Data Click this button to explicitly sync the feed permission with the selected feed data permissions. This button appears for existing feed definitions only when you select the Publish with Selected Security option.

Type Select either *Role* or *Permission List* security.

Name Enter the name of the permission list or role that has access to the feed.

Actions

Publish Click the Publish button to publish the feed definition.

Preview Feed Click to preview the published feed.

Note. This link appears for already published feed definitions only.

Publish Feed to Other Sites Click to publish the existing feed definition to additional sites.

Note. This link appears for already published feed definitions only.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Defining Advanced Feed Options

Access the Advanced Feed Options page (click the Advanced Options link on the Publish Feed Definition page).

Advanced Feed Options page (for a list of feeds type feed)

Advanced options differ by feed data type.

Max Number of Entries Enter the maximum number of entries that the feed should return to the user. Enter *0* for unlimited entries up to the maximum row limit specified on the Feed Options page, which is 300 by default.

Note. The upper limit can be configured on the Feed Options page. If the Max Number of Entries specified is either 0 or greater than the PTFP_MAX_ROW_LIMIT, then the output is limited to the value in the PTFP_MAX_ROW_LIMIT field. If PTFP_MAX_ROW_LIMIT is 0 or is undefined, then the output is limited by Max Number of Entries.

Reset to Defaults Click to reset any advanced options to their default values.

See Also

Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," Defining Advanced Options for Generic Message Feeds, page 61

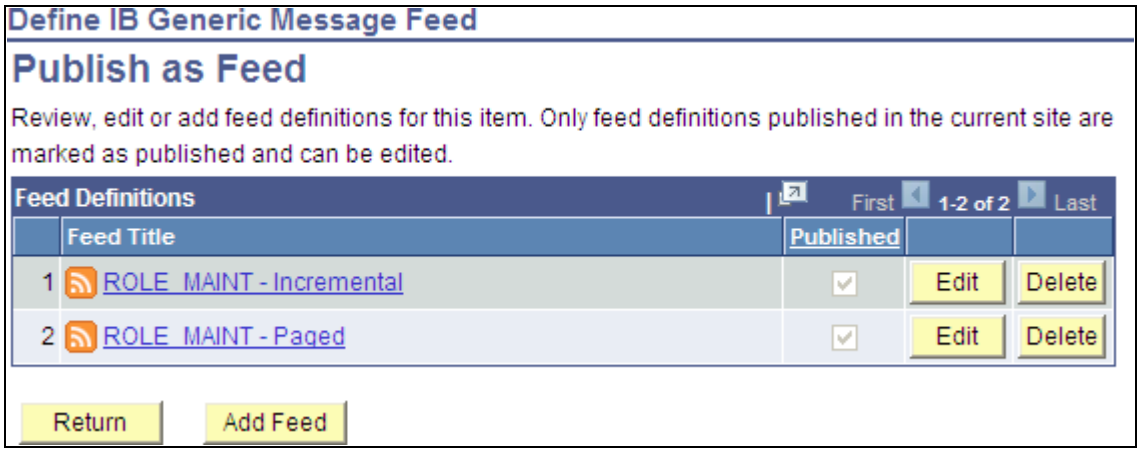
Chapter 7, "Creating and Using Query Feeds," Defining Advanced Options for Query Feeds, page 67

Chapter 8, "Creating and Using Worklist Feeds," Defining Advanced Options for Worklist Feeds, page 85

Managing Published Feeds

Access the Publish as Feed page (click the Publish as Feed link; or click the Publish button on the Publish Feed Definition page).

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Publish as Feed page (for an Integration Broker generic message type feed)

Use the Publish as Feed page to review, edit, add, or delete feed definitions for this item.

Feed Title Click a link to open the feed document for this feed definition in a separate browser window.

Edit Click this button to access the Publish Feed Definition page on which you can edit the published feed definition.

Delete Click this button to delete the feed definition.

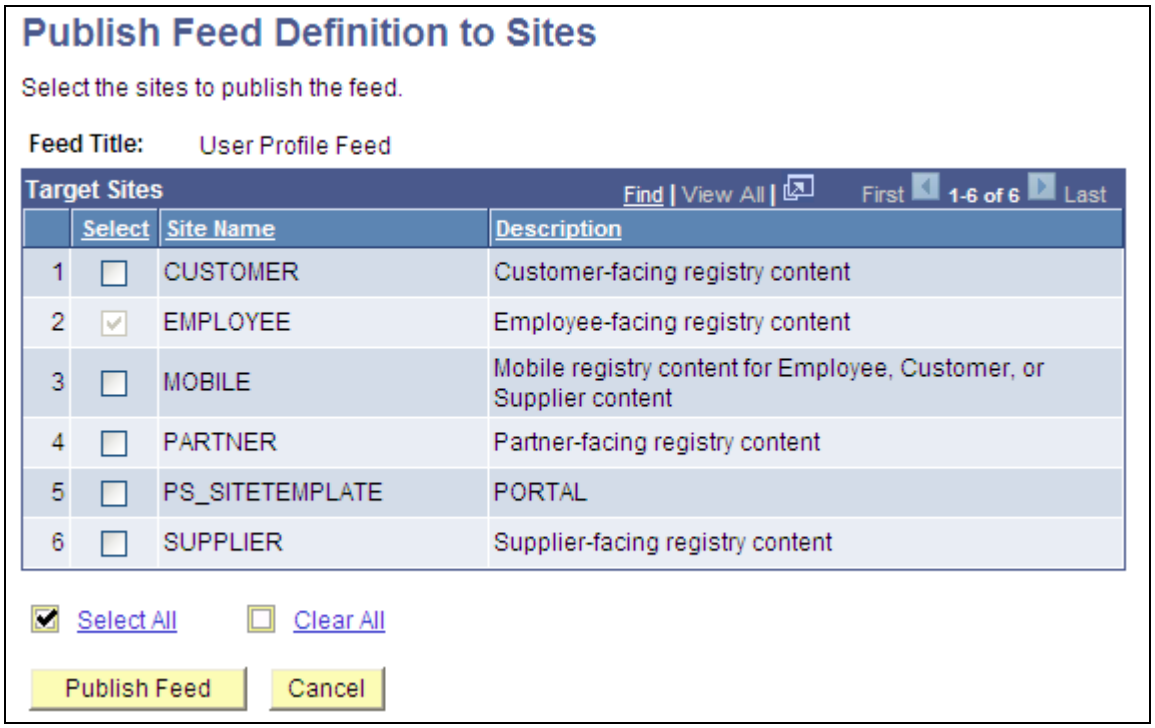
Return Click this button to return to item's administration page.

Add Feed Click this button to define and publish a new feed definition for this item.

Publishing Feeds to Additional Sites

Access the Publish Feed Definition to Sites page (click the Publish Feed to Other Sites link on the Publish Feed Definition page).

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Publish Feed Definition to Sites page

You can publish the feed to any site listed in the additional sites list.

Select Select this check box to publish the feed definition to this site.

Site Name Displays the valid sites to which you can publish the feed definition. This field is display only.

Select All Click to select all feed definitions in the list.

Clear All Click to clear the selection of any feed definitions.

Publish Feed Click to publish the feed definition to additional sites.

Publishing a List of Feeds Feed

A *list of feeds* feed is a feed that contains a list of all available feeds of a specific feed data type. When you access a list of feeds feed, you see a listing of only the feeds that you are authorized to view. You publish a list of feeds feed by using the FEED data type.

To publish a list of feeds feed:

1. Select PeopleTools, Feeds, Define Feed Data Types.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

2. Select the data type for which to produce the list of feeds:

   - *FEED*

     Select this option to produce a list of list of feeds feeds.

   - *GENERICFEED*

     Select this option to produce a list of all generic Integration Broker message feeds.

   - *PSQUERY*

     Select this option to produce a list of all query feeds.

   - *WORKLIST*

     Select this option to produce a list of all worklist feeds.

3. Click the Publish as Feed link.
4. Enter the feed properties to define the list of feeds feed.

   Note. To distinguish this as a list of feeds, you can change the feed title to include "List of" **—** for example, "List of Query Feeds".

   See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.

5. Click the Advanced Options link to specify a maximum number of entries different from the default of *10* entries.

   See Chapter 4, "Creating and Using Feeds," Defining Advanced Feed Options, page 36.

6. Click Publish to publish the feed definition.
7. Click Return to return to the Define Feed Data Types page.

Accessing Feeds

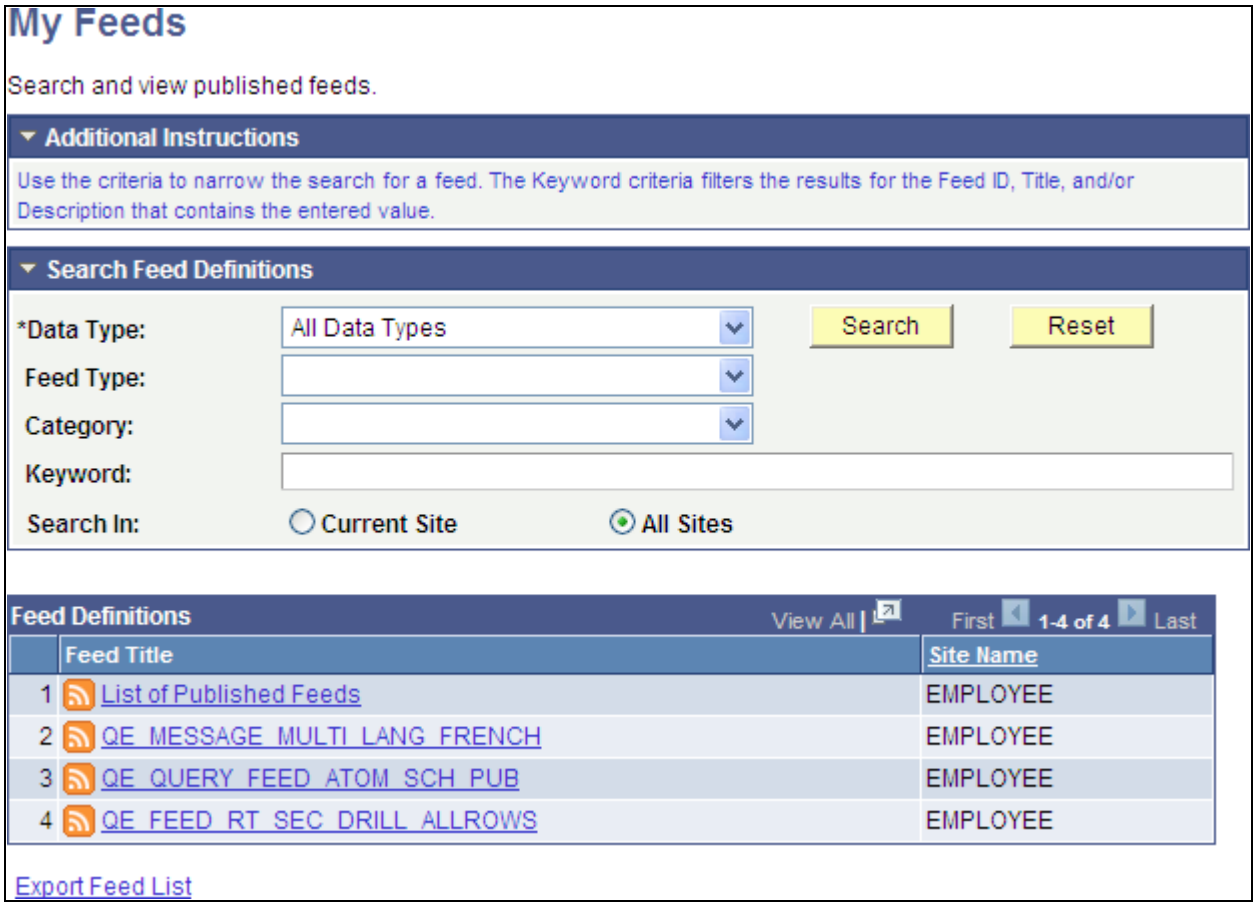
This section describes:

- Using the My Feeds page.
- Using related feeds hover menus.
- Using the GetFeedList service operation.

Using the My Feeds Page

Access the My Feeds page (click My Feeds in the Main Menu).

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



My Feeds page

Use the My Feeds page to search for and view the list of published feeds to which you have access.

Additional Instructions

The additional instructions collapsible section describes how to use the My Feeds page.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Search Feed Definitions

Data Type Select from the following feed data types:

- *All Data Types* returns a list of all feeds to which you have access.

  Note. This is the default value.

- *IB Generic Message Feeds* returns a list of Integration Broker generic message feeds.
- *List of Feeds* returns a list of all feed list feeds.
- *PS Query Feeds* returns a list of PeopleSoft Query feeds.
- *Worklist Feeds* returns a list of worklist feeds.

Note. The previous list includes only the base feed data types delivered with PeopleTools. The results can differ on your system depending which other PeopleSoft applications you have installed and whether any custom feed data types have been developed.

Feed Type Select from the following feed types:

- *Real Time* returns a list of real-time feeds.
- *Scheduled* returns a list of scheduled feeds.

Category Select from the feed categories defined on your system.

Keyword Enter a keyword to narrow the search criteria. Keywords search the *Feed ID, Feed Title,* and *Description* fields, and are *not* case sensitive.

Search In Select from the following options to search for feeds:

- *Current Site* Select this option to search in the current site only.

  Note. This is the default value.

- *All Sites* Select this option to search every site for available feeds.

Search Click the Search button to return the list of available feeds that satisfy the search criteria that you entered.

Reset Click the Reset button to clear the search results and reset all search criteria to their default values.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Export Feed List Click this link to export the current search results in the Feed Definitions grid to a file in OPML 2.0 format. OPML files can be imported in third-party feed readers, and you will be able to subscribe to each feed in the list of feeds in the exported file.

This link is visible only when a search returns one or more values.

Feed Definitions

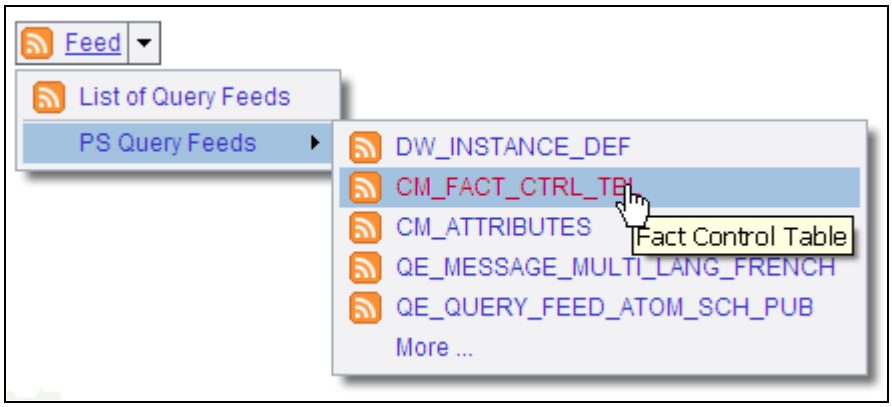
This grid lists the feeds that match the search criteria and that you have permission to access.

Note. The Site Name column is displayed when the All Sites option is selected.

Using Related Feeds Hover Menus

You can view the feeds of a particular data type to which you have access through the related feeds hover menu on a page.

Notice the hover menu extending to show the query feeds that are available to this user:



Related feeds hover menu

Note. The hover menu is inactive when no feeds have been published for that particular feed data type. Inactive hover menus can be hidden by setting a value on the Feed Options page.

See Also

Chapter 5, "Administering Feeds," Setting Feed Publishing Framework Options, page 48

Using the GetFeedList Service Operation

The GetFeedList service operation (PTFP_GETFEEDLIST) returns a list of feeds managed by the Feed Publishing Framework using the OPML format. Users can save this file to the desktop, and share the file with other users who might want to access the same list. You can also use this file to import feeds into a feed reader.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

In the file, feeds are organized in folders based on their feed data type. Only feeds visible to authenticated user of the request are returned. The Get Feed List operation provides capability to query feeds through the use of operational query parameters, the same way as the My Feeds component. Some examples of the parameters are:

Parameter	*Description*	*Example*
PTFP_DATA_TYPE	The data type being queried.	To search for the PS Query feed type, use this parameter: PTFP_DATA_TYPE=PSQUERY.
PTFP_FEED_KEYWORD	The keyword being used in the query. The keyword searches the feed title, description, and ID.	To search feeds whose name contains *User* string, use this parameter: PTFP_FEED_KEYWORD=User.
PTFP_DSS_COUNT	The data source setting name/value pair parameter count.	To specify the query string when it contains one pair of the data source setting name and value, use this parameter: PTFP_DSS_COUNT=1.
PTFP_DSS_NAMEn	The data source setting name, where n is an integer beginning with 1.	To specify the first data source setting name, use this parameter: PTFP_DSS_NAME1=QRYNAME.
PTFP_DSS_VALUEn	The data source setting value, where n is an integer beginning with 1.	To specify for the first data source setting value, use this parameter: PTFP_DSS_VALUE1=MESSAGES_FOR_MSGSET.
PORTAL_NAME	The portal being searched if you have multiple portals. The system searches all sites if this parameter is not specified.	To search only the EMPLOYEE portal, use this parameter: PORTAL_NAME=EMPLOYEE.
LANGUAGE_CD	The language of the feed being queried.	To search the feed in Canadian French, use this parameter: LANGUAGE_CD=CFR.

Note. PeopleTools provides one GetFeedList service. However, you can develop your own services for specific feed data types.

Example of .opml File Exported From the My Feeds Page

OPML refers to Outline Processor Markup Language. It is the protocol used for exchanging feed lists between feed readers and aggregators. This is an example of an opml file:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

```
<?xml version="1.0"?>
<opml version="2.0">
  <head>
    <dateCreated>Mon, 04 May 2009 16:32:27 GMT</dateCreated>
    <dateModified>Mon, 04 May 2009 16:32:27 GMT</dateModified>
    <ownerName>QE User</ownerName>
    <title>Feed Search Results</title>
  </head>
  <body>
    <outline category="IB Generic Message Feeds" description="Generic Operation"⇒
 text="IB_GENERIC" title="ADMN_IB_GENERIC" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/GetScheduled⇒
Feed?FeedID=IB_GENERIC.V1&amp;ChildFeedID=ADMN_IB_GENERIC&amp;S=P&amp;PORTAL_NAME=⇒
EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
    <outline category="PS Query Feeds" description="Dimension Control Table" text=⇒
"CM_DIM_CTRL_TBL" title="ADMN_CM_DIM_CTRL_TBL" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/RealtimeQuery⇒
Feed?FEED_ID=ADMN_CM_DIM_CTRL_TBL&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"⇒
/>
    <outline category="PS Query Feeds" description="Message Set" text="CD_MSGSET"⇒
 title="ADMN_CD_MSGSET1" type="rss" xmlUrl="http://buffy.us.oracle.com:8920/PSIGW⇒
/HttpListeningConnector/feeds/RealtimeQueryFeed?FEED_ID=ADMN_CD_MSGSET1&amp;⇒
PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
    <outline category="IB Generic Message Feeds" description="User Profile" text=⇒
"User Profile Feed" title="ADMN_USER_PROFILE" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/GetScheduled⇒
Feed?FeedID=USER_PROFILE.VERSION_84&amp;ChildFeedID=ADMN_USER_PROFILE&amp;PORTAL_⇒
NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
    <outline category="List of Feeds" description="A feed of this data type⇒
 contains data of the specified PS Query." text="PS Query Feeds" title="ADMN_PS_⇒
QUERY_FEEDS" type="rss" xmlUrl="http://buffy.us.oracle.com:8920/PSIGW/Http⇒
ListeningConnector/feeds/GetRealTimeFeed?FEED_ID=ADMN_PS_QUERY_FEEDS&amp;S=⇒
P&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
    <outline category="List of Feeds" description="A feed of this data type⇒
 contains worklist items of the current user." text="Worklist Feeds" title="ADMN_⇒
WORKLIST_FEEDS" type="rss" xmlUrl="http://buffy.us.oracle.com:8920/PSIGW/Http⇒
ListeningConnector/feeds/GetRealTimeFeed?FEED_ID=ADMN_WORKLIST_FEEDS&amp;S=⇒
P&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
    <outline category="List of Feeds" description="A feed of this data type⇒
 contains list of published feeds of the specified data type." text="List of⇒
 Feeds" title="ADMN_LIST_OF_FEEDS" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/GetRealTimeFeed?⇒
FEED_ID=ADMN_LIST_OF_FEEDS&amp;S=P&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_⇒
LOCAL"/>
    <outline category="PS Query Feeds" description="User ID's Access Activity"⇒
 text="User Access Activity" title="ADMN_PT_SEC_ACCESSLOG_USER" type="rss" xmlUrl=⇒
"http://buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/Realtime⇒
QueryFeed?FEED_ID=ADMN_PT_SEC_ACCESSLOG_USER&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_⇒
NAME=QE_LOCAL"/>
    <outline category="PS Query Feeds" description="Hierarchy Map Table" text="CM_⇒
HIER_MAP_TBL" title="ADMN_CM_HIER_MAP_TBL" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/RealtimeQuery⇒
Feed?FEED_ID=ADMN_CM_HIER_MAP_TBL&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"⇒
/>
```


Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

```
    <outline category="PS Query Feeds" description="Attribute mappings" text="CM_⇒
ATTRIBUTES" title="ADMN_CM_ATTRIBUTES" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/RealtimeQuery⇒
Feed?FEED_ID=ADMN_CM_ATTRIBUTES&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
    <outline category="Worklist Feeds" description="Worklist notifications" text=⇒
"Worklist Notification" title="ADMN_WORKLIST_ITEMS" type="rss" xmlUrl="http:⇒
//buffy.us.oracle.com:8920/PSIGW/HttpListeningConnector/feeds/GetRealTimeFeed?⇒
FEED_ID=ADMN_WORKLIST_ITEMS&amp;PORTAL_NAME=EMPLOYEE&amp;NODE_NAME=QE_LOCAL"/>
  </body>
</opml>
```

Viewing Feeds

You can view feeds directly in the browser or by using a third-party feed reader program.

To view feeds:

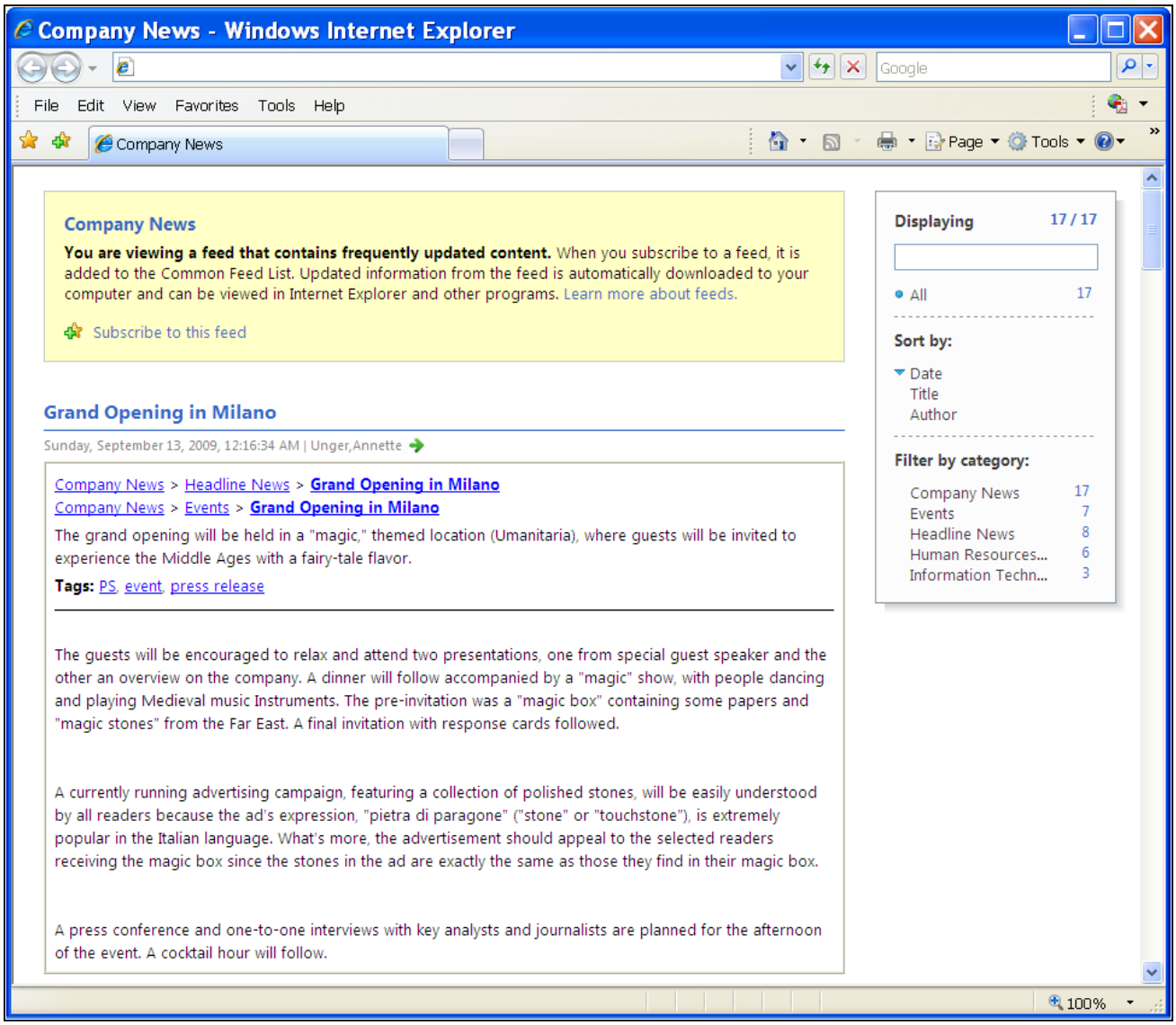
1. Click the feed link.

   The feed will open in the browser.

2. Alternatively, paste the URL into a feed reader.

This is an example of a feed in the browser:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Company News feed in the browser

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 5

Administering Feeds

This chapter discusses how to:

- Administer the Feed Publishing Framework.
- Migrate feeds between databases.
- Use SysAudit information.
- Archive feeds.

Administering the Feed Publishing Framework

This section discusses how to:

- Set Feed Publishing Framework options
- Define feed categories
- Copy feed definitions
- Delete feed definitions

Pages Used to Administer the Feed Publishing Framework

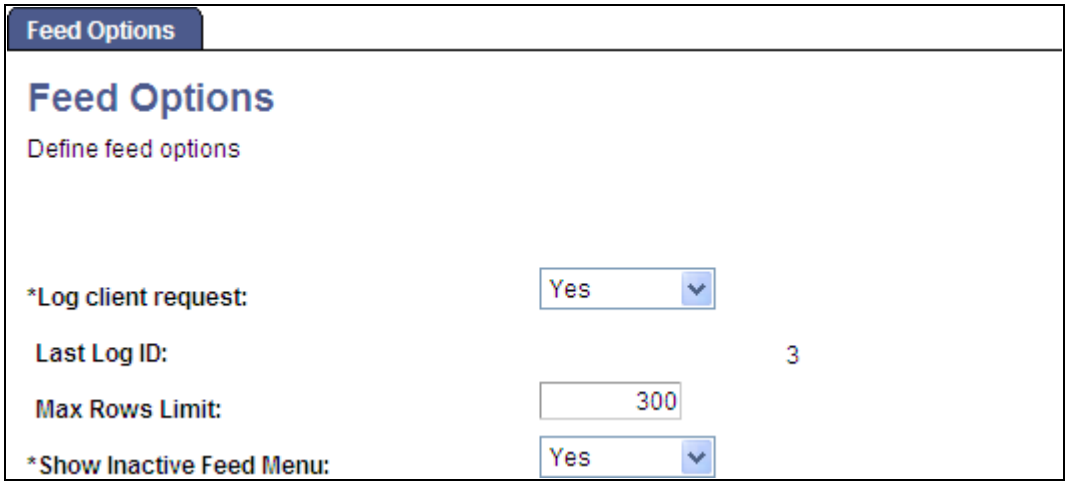
Page Name	*Definition Name*	*Navigation*	*Usage*
Feed Options	PTFP_OPTIONS	PeopleTools, Feeds, Feed Options	Define system-wide options for the Feed Publishing Framework.
Define Feed Categories	PTFP_CATEGORIES	PeopleTools, Feeds, Define Feed Categories	Manage the categories used to organize feeds.
Copy Feed Definitions	PTFP_SAVEAS	PeopleTools, Feeds, Copy Feed Definitions	Clone an existing feed definition to one with a new feed ID.
Delete Feed Definitions	PTFP_DELETE	PeopleTools, Feeds, Delete Feed Definitions	Delete the selected feed definitions from all sites.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Page Name	Definition Name	Navigation	Usage
Define Feed Categories - Delete Confirmation	PTFP_CONFIRM_DEL	Click the Delete Category button on the Define Feed Categories page.	Confirm the deletion of a feed category.
Delete selected feeds? – Delete Confirmation	PTFP_CONFIRM_DEL	Click the Delete Selected Feeds button on the Delete Feed Definitions page.	Confirm the deletion of a feed definition.
Copy Feed Definitions – Copy confirmed	PTFP_CONFIRM_SAVE	Click the Save Selected Feed button on the Copy Feed Definitions page.	Confirm the copy of a feed definition.

Setting Feed Publishing Framework Options

Access the Feed Options page (PeopleTools, Feeds, Feed Options).



Feed Options page

Use the Feed Options page to define system-wide options for the Feed Publishing Framework.

Log client request Select *Yes* to enable logging of feed requests in the PTFP_ACCESS_LOG table. The default value is *No.*

Last Log ID Displays the log ID of the last feed request.

Max Rows Limit Specify the limit for the maximum number of feed entries in a feed. This limit applies to all feed types except paged scheduled feeds.

Show Inactive Feed Menu Select *No* to hide the related feeds hover menu when it is inactive—that is, when no feeds of that type have been defined. The default value is *Yes,* show the related feeds hover menu when it is inactive.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Defining Feed Categories

Access the Define Feed Categories page (PeopleTools, Feeds, Define Feed Categories).

Define Feed Categories page

Use the Define Feed Categories page to manage categories used to organize feeds.

Category ID Specify a unique ID for the feed category.

Note. The category ID is a user-specified value and not system defined.

Description Enter a short description for the feed category.

This description displays wherever a category can be assigned to a feed definition and in locations where you can search for feeds.

Active Indicate whether the feed category is active or inactive.

An inactive category is not available to assign to feed definitions or in locations where you can search for feeds. However, if a feed definition is assigned to a category and then that category is made inactive, the feed definition no longer displays an assigned category.

Long Description Enter a long description for the feed category.

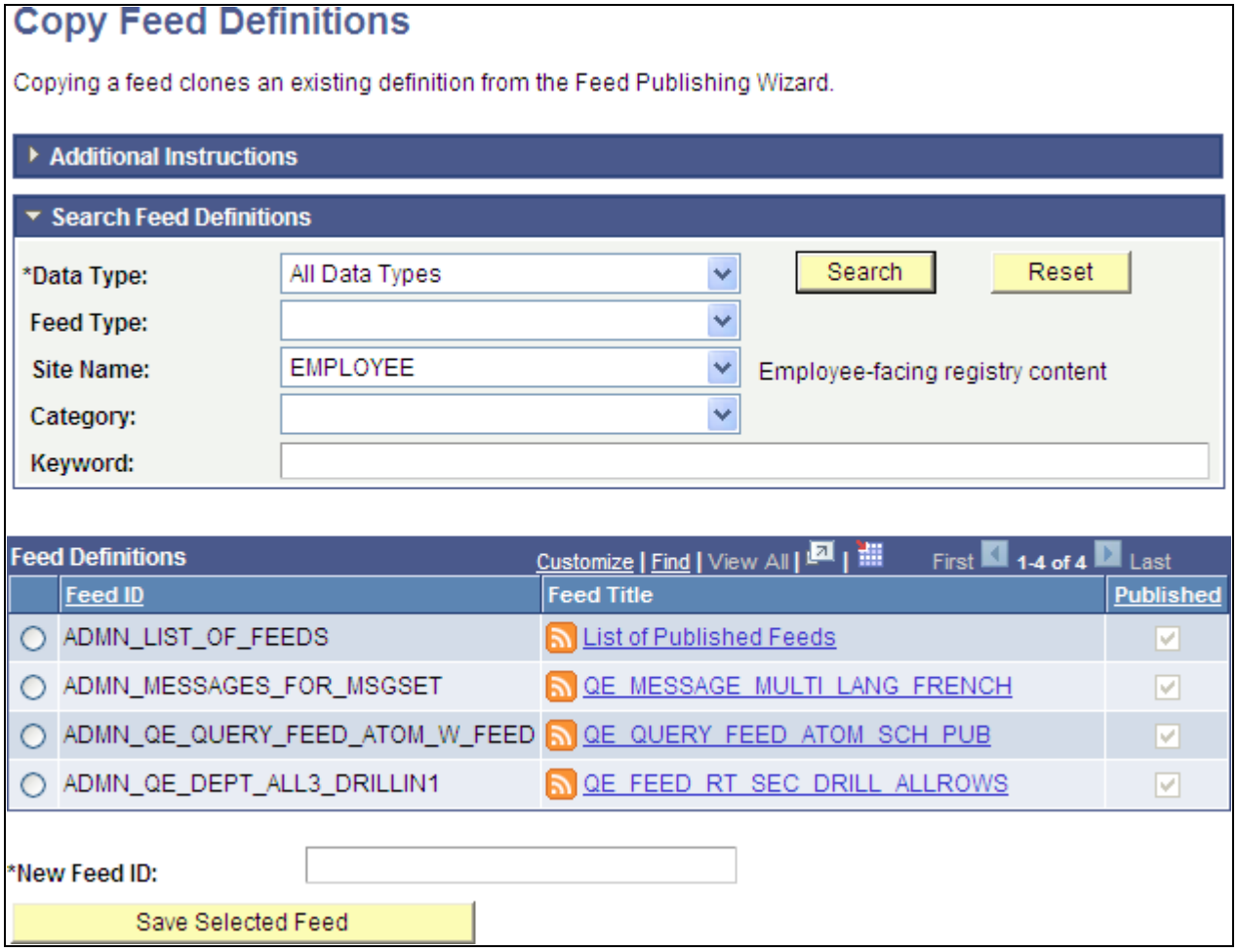
Delete Category Click to delete the feed category.

If a feed definition is assigned to a category and then that category is deleted, the feed definition no longer displays an assigned category.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Copying Feed Definitions

Access the Define Feed Categories page (PeopleTools, Feeds, Copy Feed Definitions).



Copy Feed Definitions page

Use the Copy Feed Definitions page to clone an existing feed definition to one with a new feed ID. The new feed definition is published witht the same feed options and to the same sites as the original.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Data Type

Select from the following feed data types:

- *All Data Types* returns a list of all feeds to which you have access.

  Note. This is the default value.

- *IB Generic Message Feeds* returns a list of Integration Broker generic message feeds.
- *List of Feeds* returns a list of all feed list feeds.
- *PS Query Feeds* returns a list of PeopleSoft Query feeds.
- *Worklist Feeds* returns a list of worklist feeds.

Note. The previous list includes only the base feed data types delivered with PeopleTools. The results can differ on your system depending which other PeopleSoft applications you have installed and whether any custom feed data types have been developed.

Feed Type

Select from the following feed types:

- *Real Time* returns a list of real-time feeds.
- *Scheduled* returns a list of scheduled feeds.

Site Name

Select from the sites available on your system.

This field does not filter the search results; the results indicate whether a feed definition is published to the selected site.

Category

Select from the active feed categories defined on your system.

Keyword

Enter a keyword to narrow the search criteria. Keywords search the *Feed ID, Feed Title,* and *Description* fields, and are *not* case sensitive.

Search

Click the Search button to return the list of available feeds that satisfy the search criteria that you entered.

Reset

Click the Reset button to clear the search results and reset all search criteria to their default values.

Feed Definitions

Select the feed from this grid that you wish to copy.

Note. The Published field indicates that the feed definition is published to the site selected in the Site Name field.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

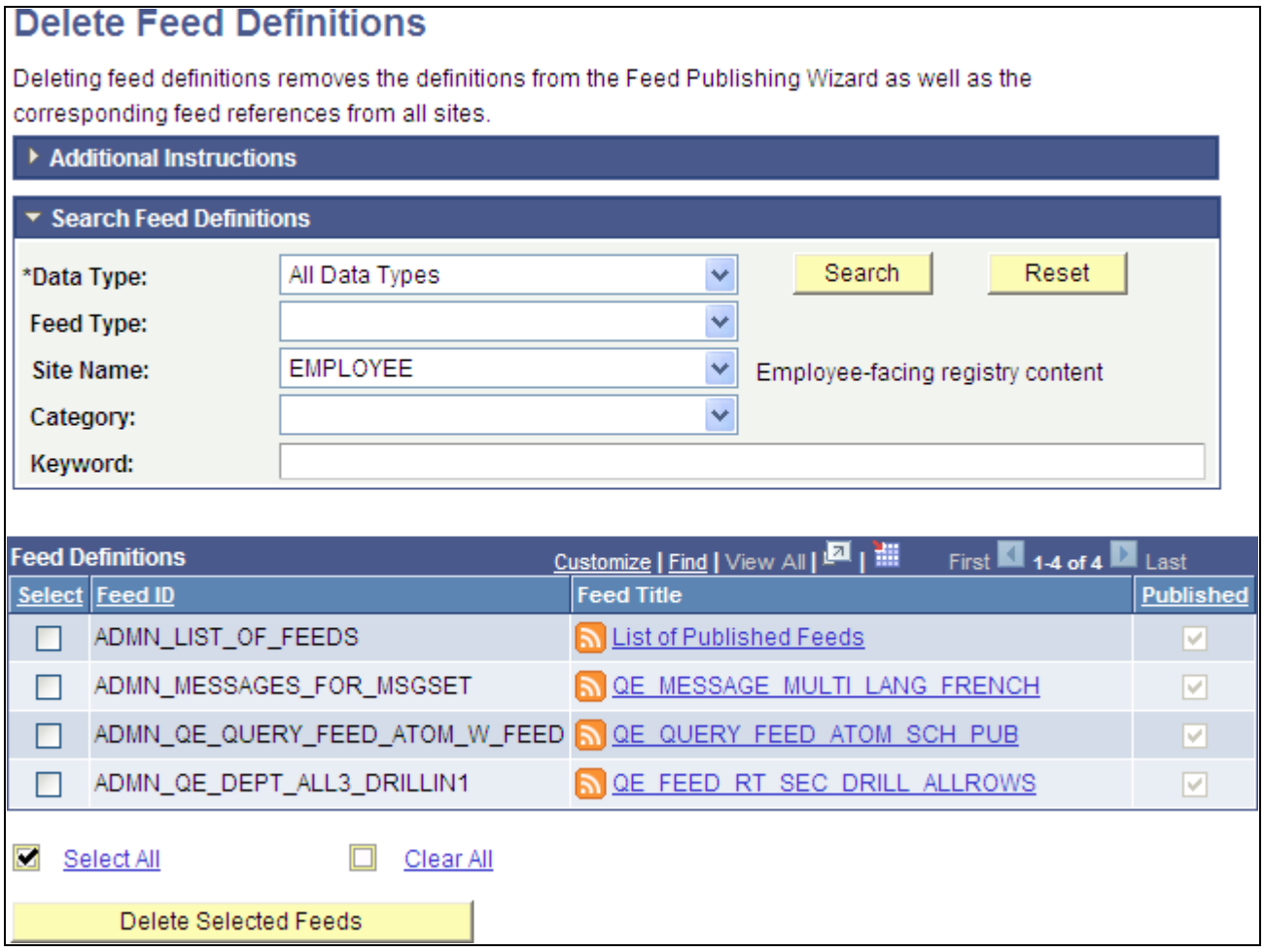
New Feed ID Enter an ID for the new feed definition with alphabetic characters (A–Z), numeric characters (0–9), and the underscore character (_) only.

Note. All lowercase characters will be converted to uppercase by the system. In addition, any other special characters such as a space, #, @, and so on will be converted to an underscore by the system.

Save Selected Feed Click to copy the selected feed definition to a new feed definition.

Deleting Feed Definitions

Access the Delete Feed Definitions page (PeopleTools, Feeds, Delete Feed Definitions).



Delete Feed Definitions page

Use the Delete Feed Definitions page to delete the selected feed definitions. The feed definition is deleted from all sites to which it has been published.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Data Type

Select from the following feed data types:

- *All Data Types* returns a list of all feeds to which you have access.

  Note. This is the default value.

- *IB Generic Message Feeds* returns a list of Integration Broker generic message feeds.
- *List of Feeds* returns a list of all feed list feeds.
- *PS Query Feeds* returns a list of PeopleSoft Query feeds.
- *Worklist Feeds* returns a list of worklist feeds.

Note. The previous list includes only the base feed data types delivered with PeopleTools. The results can differ on your system depending which other PeopleSoft applications you have installed and whether any custom feed data types have been developed.

Feed Type

Select from the following feed types:

- *Real Time* returns a list of real-time feeds.
- *Scheduled* returns a list of scheduled feeds.

Site Name

Select from the sites available on your system.

This field does not filter the search results; the results indicate whether a feed definition is published to the selected site.

Category

Select from the feed categories defined on your system.

Keyword

Enter a keyword to narrow the search criteria. Keywords search the *Feed ID, Feed Title,* and *Description* fields, and are *not* case sensitive.

Search

Click the Search button to return the list of available feeds that satisfy the search criteria that you entered.

Reset

Click the Reset button to clear the search results and reset all search criteria to their default values.

Feed Definitions

Select the feed definition or definitions from this grid that you want to delete.

Note. The Published field indicates that the feed definition is published to the site selected in the Site Name field.

Select All

Click to select all feed definitions in the list.

Clear All

Click to clear the selection of any feed definitions.

Delete Selected Feeds

Click to delete the selected feed definition or definitions.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Migrating Feeds Between Databases

Use Data Mover to migrate feeds between databases.

See *PeopleTools 8.51 PeopleBook: Data Management*, "Using PeopleSoft Data Mover."

This section discusses how to migrate between databases.

Moving Existing Seed Data

To move existing seed data:

1. To export existing feed seed data from the source database, in the *source* database open Data Mover and then modify and run the ptfp_setup_exp.dms script.
2. To import existing feed seed data into the target database, in the *target* database open Data Mover and then modify and run the ptfp_setup_imp.dms script.

Exporting Feed Definitions

To export feed definitions, use the script that is provided as message catalog entry: message set 219, message 3300 (without user personalization) or message 3302 (with user personalization) . To export feed definitions, perform the following tasks:

1. Select PeopleTools, Utilities, Administration, Message Catalog.
2. Enter *219* in the Message Set Number field and press the Enter key.
3. Click the Find link and enter *3300* in the Enter Search String field.

   This will retrieve the script for exporting feed definitions without user personalization. Enter *3302* to export with user personalization.
4. Click the OK button.
5. Copy the script from the Description field.
6. Open Data Mover in the *source* database.
7. Paste the script into the upper pane, modify the script as necessary, and run the script.

Importing Feed Definitions

To import feed definitions, use the script that is provided as message catalog entry: message set 219, message 3301 (if you have exported the feed definition without user personalization) or 3303 (if you have exported the feed definition with user personalization). To import feed definitions, perform the following tasks:

1. Select PeopleTools, Utilities, Administration, Message Catalog.
2. Enter *219* in the Message Set Number field and press the Enter key.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

3. Click the Find link and enter *3301* in the Enter Search String field.

   This will retrieve the script for importing feed definitions without user personalization. Enter *3303* to import with user personalization.

4. Click the OK button.

5. Copy the script from the Description field.

6. Open Data Mover in the *target* database.

7. Paste the script into the upper pane, modify the script as necessary, and run the script.

Note. The import script deletes any feeds with the same feed ID in the target database before importing the feed definitions.

See Also

PeopleTools 8.51 PeopleBook: Data Management, "Using PeopleSoft Data Mover," Running Scripts

Using SysAudit Information

SysAudit information is available for:

- Feed definition integrity.
- PS Query feed integrity.
- Worklist feed integrity.

See Also

PeopleTools 8.51 PeopleBook: Data Management, "Ensuring Data Integrity," Running SYSAUDIT

Archiving Feeds

This section provides an overview of archiving feed data and discusses how to archive scheduled feed data.

Understanding the Archiving of Feed Data

Scheduled feeds are published asynchronously and stored as messages in Integration Broker queues. The process of archiving feed data archives the messages stored in the Integration Broker queues used by scheduled feeds. You run a batch process to archive and delete the queue message data.

This section provides an overview of:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Feed archiving parameters.
- Feed archiving options and logic.

Feed Archiving Parameters

The following parameters specified in the feed definition are used for archiving the Integration Broker queue messages:

- Integration Broker Service Operation name

  For generic scheduled feeds, the service operation name is the value of the IB_OPERATIONNAME data source setting.

  For other scheduled feeds, the system uses the service operations listed in the Feed Service Operations grid on the Define Data Types page.

- Integration Broker subqueue name, when applicable

  For generic scheduled feeds, no subqueue name exists.

  For other scheduled feeds, the Feed ID value is the sub queue name.

- PTFP_FEED:UTILITY:Utility DSPARAMETER_MAXROW

  For scheduled feeds, this parameter must be defined in the data source parameter for the data type.

  This parameter indicates the number of feed items that will appear to the user.

  - If the parameter value is *PTFP_FEED:UTILITY:Utility SF_MAXROWOPTION_LATESTMSG*, then only the latest message in the Integration Broker queue appears to the user.
  - If the parameter value is *PTFP_FEED:UTILITY:Utility SF_MAXROWOPTION_ALLMSGS*, then all the messages in the Integration Broker queue appear to the user.
  - The parameter can be a number, for example, *4*, which indicates that only 4 items appear to the user.

- PTFP_FEED:UTILITY:Utility DSPARAMETER_SF_MAXMINUTES

  For scheduled feeds, this parameter must be defined in the data source parameter for the data type.

  This parameter tells the length of time for which the feed messages are valid in the Integration Broker queues.

  - If the parameter value is *PTFP_FEED:UTILITY:Utility SF_MAXMINUTES_ALLMSGS*, then all the messages appear to the user.
  - If the parameter value is a number, for example *100,* then all the feed messages existing in the Integration Broker queue for fewer than 100 minutes appear to the user.

Feed Archiving Options and Logic

This table describes the feeds archiving options and logic:

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Option Value	Description	Archiving Logic
0	Archive all messages that are not within the specified date and time range.	When DSPARAMETER_SF_MAXMINUTES is not SF_MAXMINUTES_ALLMSGS.
1	Archive all messages except latest 1 message per subqueue per language.	When DSPARAMETER_SF_MAXMINUTES is SF_MAXMINUTES_ALLMSGS and DSPARAMETER_MAXROW is SF_MAXROWOPTION_LATESTMSG.
2	Archive all messages except latest *n* messages per subqueue per language.	When DSPARAMETER_SF_MAXMINUTES is SF_MAXMINUTES_ALLMSGS and DSPARAMETER_MAXROW is some number. This number indicates the number of messages that should *not* be archived.
3	Archive nothing.	When DSPARAMETER_SF_MAXMINUTES is SF_MAXMINUTES_ALLMSGS and DSPARAMETER_MAXROW is SF_MAXROWOPTION_ALLMSGS.

Note. The feed archiving options are implicitly arrived at, based on the feed definition. These cannot be specified when the PTIBFEEDARCH process runs.

Pages Used to Archive Feeds

Page Name	Definition Name	Navigation	Usage
Run Feed Archive	IB_FEEDARCHIVE	PeopleTools, Feeds, Archive Feed Data	Archive the messages stored in the Integration Broker queues used by scheduled feeds. Run a batch process to archive and delete the queue message data.

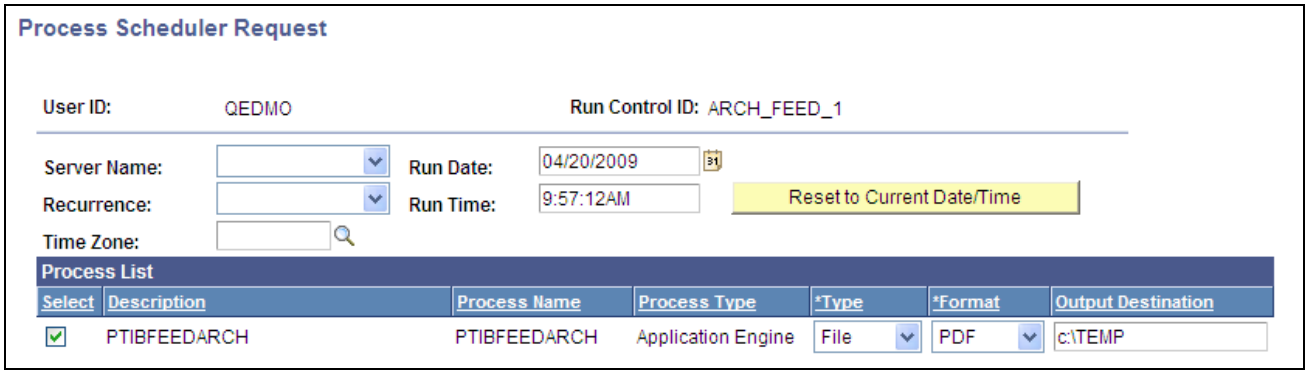
Archiving Scheduled Feed Data

Access the Run Feed Archive page (PeopleTools, Feeds, Archive Feed Data).

Run Feed Archive page

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Click the Run button to access the PTIBFEEDARCH process:



Process Scheduler Request page showing the PTIBFEEDARCH process

Use the Run Feed Archive page to archive Integration Broker messages that are used by scheduled feeds. Archiving takes no parameters from the user; parameters for archiving come from the feed definitions themselves.

All scheduled feeds are archived simultaneously; you cannot selectively archive feeds. The process archives and deletes the feed data based on the Archive flag in the Integration Broker queue used in the service operations that published messages to Integration Broker queues. If the Archive flag is enabled, then the messages are archived then deleted; otherwise, the messages are just deleted.

You can view archived Integration Broker messages by using the Service Operations Monitor. Look for asynchronous services with the Archive flag enabled.

Note. A scheduled feed can be archived only if the service operations involved in the scheduled feeds are listed in the Feed Service Operations grid on the Define Feed Data Types page.

Note. Feeds archiving logic is comparable to Integration Broker archiving logic. For a feed message to be archived, you must select the Archive check box on the Queue Definition page for the service operation. If you do not select the Archive check box, then the feeds archiving process deletes the feed messages and the messages do not appear on the Service Operation Monitor - Asynchronous Services page when the Archive check box is selected.

See *PeopleTools 8.51 PeopleBook: Integration Broker Service Operations Monitor*, "Monitoring Asynchronous Service Operations," Monitoring Asynchronous Service Operation Transactions.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 6

Creating and Using Integration Broker Generic Message Feeds

This chapter provides an overview of Integration Broker generic message feeds and discusses how to:

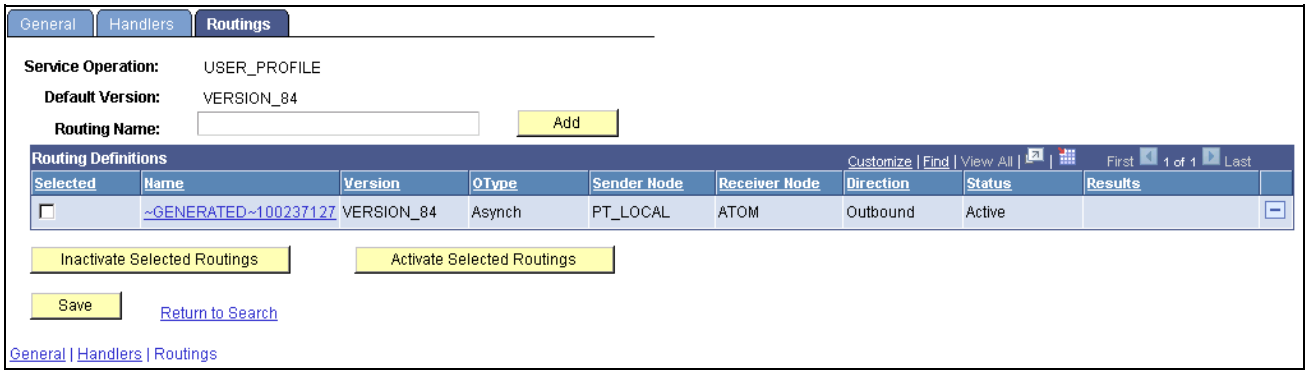
- Publish Integration Broker generic message feeds.
- Use Integration Broker generic message feeds.

Understanding Integration Broker Generic Message Feeds

Integration Broker generic message feeds enable administrators to expose Integration Broker messages used in asynchronous, one-way service operations as feeds. These feeds are scheduled feeds. Unlike up-front feeds, where the message contents in Integration Broker queues are feed messages, the messages published to Integration Broker queues in the case of generic feeds are either PeopleSoft rowset messages or non-rowset messages. These messages are called Integration Broker generic messages. Typically, these messages are generated based on an actions performed by users or events occurring within the PeopleSoft system.

The Define Generic Feed page allows the feed administrator to select any existing asynchronous, one-way service operation in the system and expose it as a feed. When a feed is exposed for an asynchronous, one-way operation, the framework automatically creates a Local-to-Atom routing for the corresponding service operation.

This example of the Routings page shows the Local-to-Atom routing for the USER_PROFILE service operation:



Example of service operation showing Local-to-Atom routing

When a message is published for an asynchronous, one-way service operation, these messages will also be available in the feed, which is another form of application integration.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

The default and recommended security option for generic Integration Broker message feeds is real-time security, although you can select other options as necessary. Real-time security allows only users who are in the permission list on the service operation's security page to access that Integration Broker generic message as a feed. Users in the PeopleSoft Administrator and Portal Administrator roles also have access to Integration Broker generic message feeds.

Publishing Integration Broker Generic Message Feeds

This section provides an overview of the steps used to publish an Integration Broker generic message feed and discusses how to:

- Publish an Integration Broker generic message as a feed.
- Define advanced options for generic message feeds.

Understanding the Steps to Publish an Integration Broker Generic Message Feed

To publish Integration Broker generic message feeds:

1. Select PeopleTools, Feeds, Define IB Generic Message Feed.
2. Select the service operation to publish as a feed.
3. On the Define IB Generic Message Feed page, click the Publish as Feed link.

   See Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," Publishing an Integration Broker Generic Message as a Feed, page 61.
4. On the Publish Feed Definition page, define the feed properties and security.

   See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.
5. Click the Advanced Options link.
6. Define the advanced feed options.

   See Chapter 6, "Creating and Using Integration Broker Generic Message Feeds," Defining Advanced Options for Generic Message Feeds, page 61.
7. Click the OK button.
8. Click the Publish button.

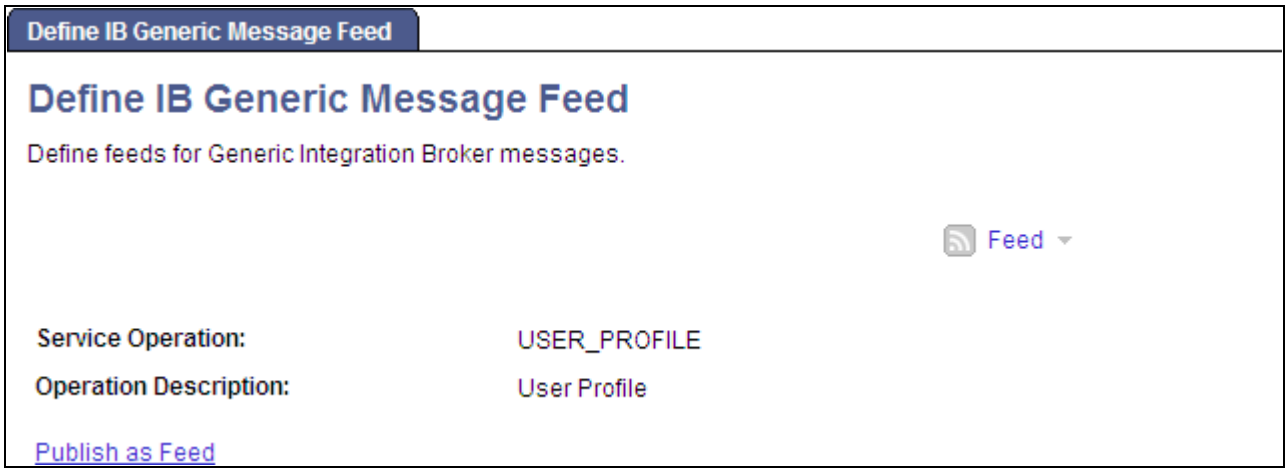
 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Pages Used to Publish Integration Broker Generic Message Feeds

Page Name	*Definition Name*	*Navigation*	*Usage*
Define IB Generic Message Feed	PTFP_GENERIC_FEED	PeopleTools, Feeds, Define IB Generic Message Feed	Create feed definitions for generic Integration Broker asynchronous, one-way service operations.
Define IB Generic Message Feed - Advanced Feed Options	PTFP_GENFD_ADVOPT	Click the Advanced Options link on the Publish Feed Definition page.	Enter advanced option values that are specific to generic message feeds.

Publishing an Integration Broker Generic Message as a Feed

Access the Define IB Generic Message Feed page (PeopleTools, Feeds, Define IB Generic Message Feed).



Define IB Generic Message Feed page

Use the Define IB Generic Message Feed page to publish an Integration Broker asynchronous, one-way service operation as a feed. Click the Publish as Feed link to begin creating the feed definition.

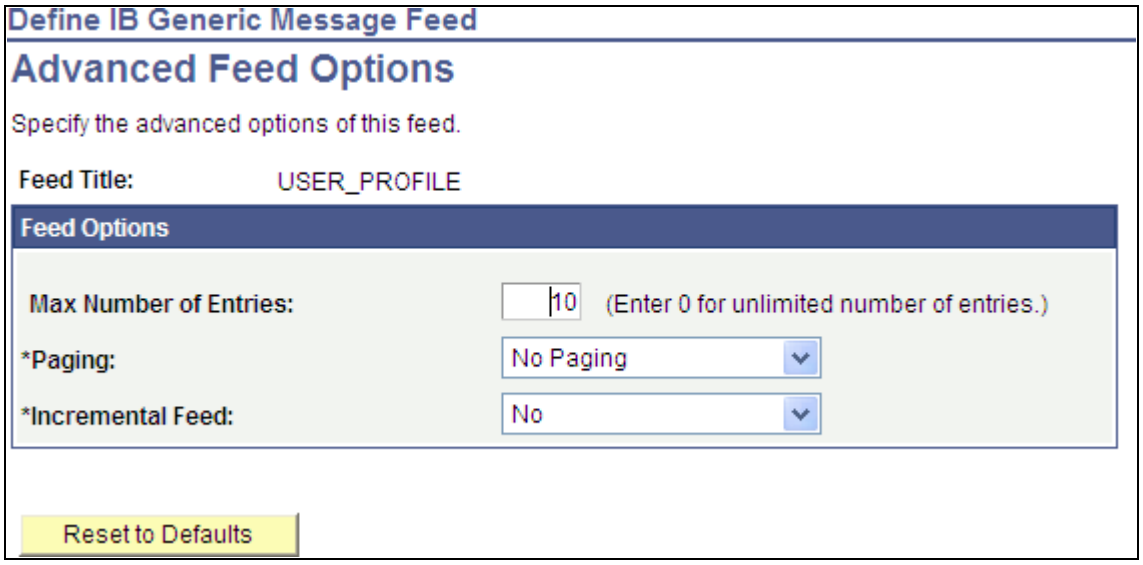
Create the feed definition on the Publish Feed Definition page.

See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.

Defining Advanced Options for Generic Message Feeds

Access the Advanced Feed Options page (click the Advanced Options link on the Publish Feed Definition page).

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Define IB Generic Message Feed - Advanced Feed Options page

Use the Advanced Feed Options page to define advanced options for Integration Broker generic message feeds.

Max Number of Entries Enter the maximum number of entries that the feed should return to the user. Enter *0* for unlimited entries up to the maximum row limit specified on the Feed Options page, which is 300 by default.

Note. The upper limit can be configured on the Feed Options page.

Paging Select one of these options:

- *No Paging* – Indicates that the feed is not paged. This is the default value.
- *Segmented* – Indicates that the feed is paged.

  A *paged feed* is a feed that has been split into pages (also know as segments) to improve system performance in delivering the feed document and to improve performance for consuming a feed. A paged feed is presented with first, last, next, and previous links to provide access to additional pages in the feed document.

  Note. When *Segmented* is selected as the paging option, then the Max Number of Entries and Incremental Feed fields are disabled.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Incremental Feed Select one of these options:

- *Yes* – Indicates that the feed is incremental.

  An *incremental feed* is a feed that has been published and updated with timestamps that allow the feed content to be delivered incrementally. An incremental feed allows the Feed Publishing Framework to deliver only the feed content that has changed since the feed was last requested by the user.

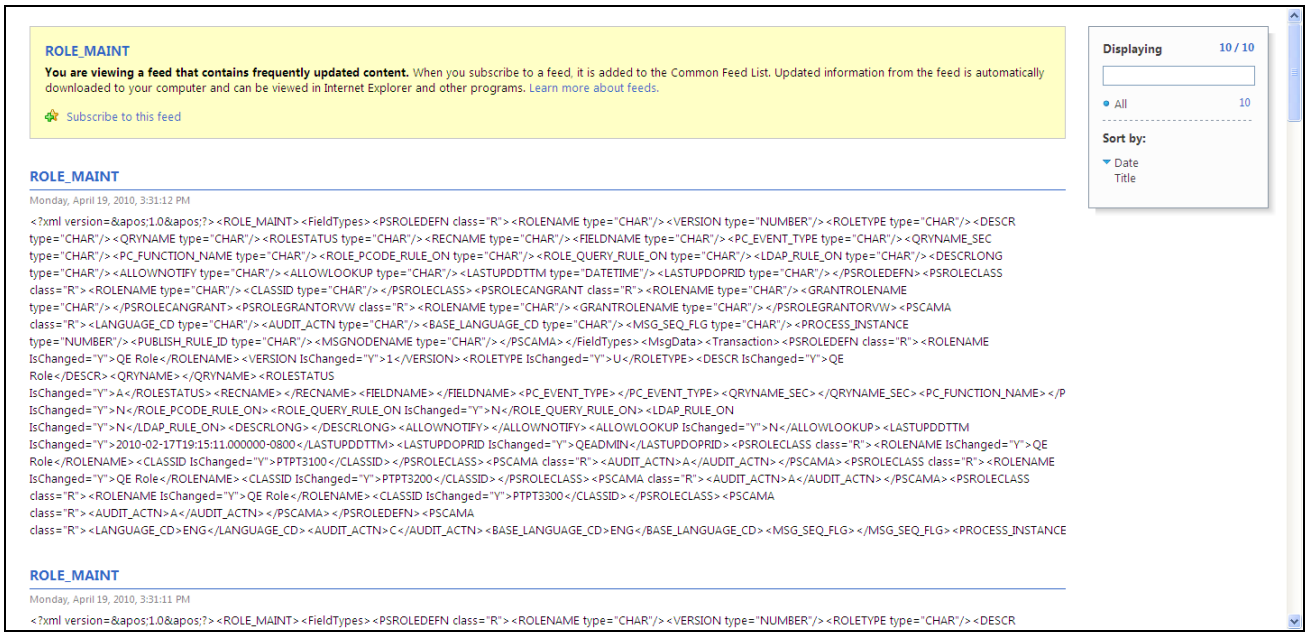
- *No*– Indicates that the feed is incremental. This is the default value.

Reset to Defaults Click to reset the advanced options to their default values.

Using Integration Broker Generic Message Feeds

Integration Broker generic message feeds enable administrators to expose Integration Broker messages used in asynchronous, one-way service operations as feeds.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Generic message feed for the ROLE_MAINT service operation

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 7

Creating and Using Query Feeds

This chapter provides an overview of query feeds and discusses how to:

- Publish query feeds.
- Define advanced options for query feeds.
- Use query feeds.

Understanding Query Feeds

This section discusses query feeds.

Query Feed Security

There are two levels of security in the Feed Publishing Framework: feed security and data security. Feed security controls who can see the feed when searching for feeds. Data security controls whether the underlying data can be seen in the feed entries when a user requests the feed document.

Query feed security, which feed administrators specify on the advanced options page, determines feed security—that is, who can see the feed in search results.

Publishing Query Feeds

This section provides an overview of the steps used to publish a query feed and discusses how to publish a query as a feed.

Understanding the Steps to Publish a Query Feed

To publish a query as a feed:

Note. Any user with access to Query Manager can publish query feeds.

1. Select Reporting Tools, Query, Query Manager.
2. Select the query that you want to publish as a feed and click the Edit link.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

3. Click the Publish as Feed link.

   Note. If a feed has already been published for this query, then the link is displayed as Manage Feeds rather than Publish as Feed.

   See Chapter 7, "Creating and Using Query Feeds," Publishing a Query as a Feed, page 66.

4. On the Publish Feed Definition page, define the feed properties.

   See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.

5. Click the Advanced Options link.

6. Enter the query parameters, define advanced options such as feed security, and map feed entry elements for the query feed.

   See Chapter 7, "Creating and Using Query Feeds," Defining Advanced Options for Query Feeds, page 67.

7. Click the OK button.

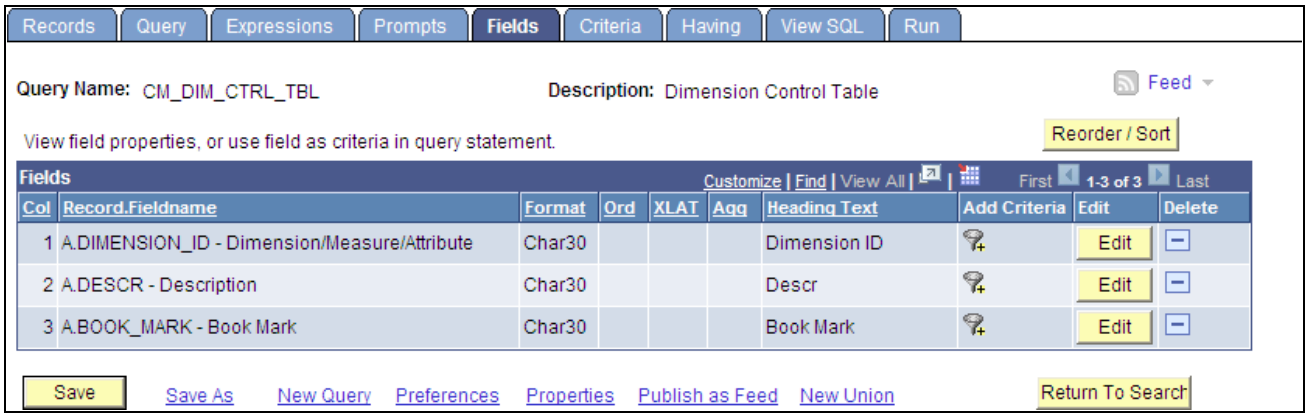
8. Click the Publish button.

Pages Used to Publish Query Feeds

Page Name	*Definition Name*	*Navigation*	*Usage*
Fields	QRY_FIELDS	Reporting Tools, Query, Query Manager	Create feed definitions for queries.

Publishing a Query as a Feed

Access the Fields page (Reporting Tools, Query, Query Manager).



Fields page

Use the Fields page to publish a query as a feed. Click the Publish as Feed link to begin creating the feed definition.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Note. If a feed has already been published for this query, then the link is displayed as Manage Feeds rather than Publish as Feed.

Create the feed definition on the Publish Feed Definition page.

See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.

Defining Advanced Options for Query Feeds

This section discusses how to:

- Enter advanced feed options for query feeds.
- Map feed entry elements to entry templates.
- Use Mapping Builder to edit entry templates.

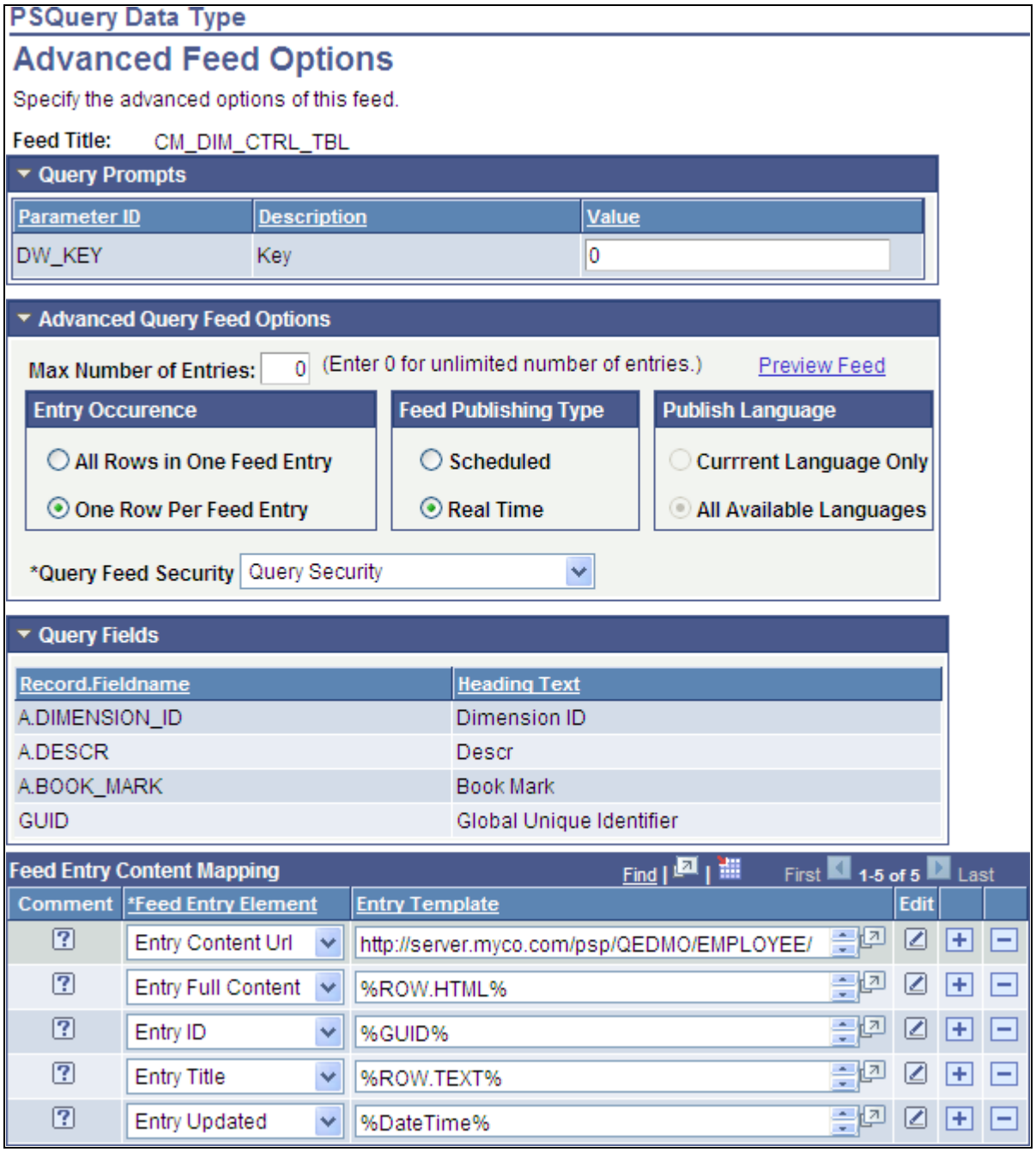
Pages Used to Define Advanced Feed Options for Query Feeds

Page Name	*Definition Name*	*Navigation*	*Usage*
PSQuery Data Type - Advanced Feed Options	PTPSQRY_ADV_OPT	Click the Advanced Options link on Publish Feed Definition page.	Enter the query parameters, define advanced options such as feed security, and map feed entry elements for the query feed.
Feed Element Mapping Builder	PTPSQRY_MAP_ELMT	Click the Edit button on the Feed Entry Content Mapping grid.	Create entry templates to customize the output of the elements that comprise each feed entry of a query feed.

Enter Advanced Feed Options for Query Feeds

Access the Advanced Feed Options page (click the Advanced Options link on the Publish Feed Definition page).

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



PSQuery Data Type - Advanced Feed Options page

Query Prompts

Any fields that contain prompts appear in this area. Enter the prompt values that the query should return before you publish the feed. If you do not enter necessary prompt values, then the published feed contains no entries.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Advanced Query Feed Options

Max Number of Entries Enter the maximum number of entries that the feed should return to the user. Enter *0* for unlimited entries up to the maximum row limit specified on the Feed Options page, which is 300 by default.

Note. The upper limit can be configured on the Feed Options page.

Preview Feed Click to save and temporarily publish and view the feed.

Entry Occurrence Select from these options:

All Rows in One Feed Entry: Select this option to have the feed reader display one entry for the entire query.

One Row Per Feed Entry: Select this option to have the feed reader display one entry for each row returned by the query.

Feed Publishing Type Select from these options:

Scheduled: Select this option to use Query Scheduler to schedule the query to run once and be stored in the Integration Broker queues. This option provides superior performance, but it should be used when the data changes infrequently. If you select this option, you may select whether to publish the feed in the language of the current user or all available languages.

See *PeopleTools 8.51 PeopleBook: PeopleSoft Query*, "Modifying, Scheduling, and Organizing Queries," Scheduling Queries.

Real Time: Select this option to run the query in real-time each time a user requests that this query feed be updated in the feed reader. If you select this option, you can only publish in the language of the user configuring the feed.

Publish Language Active only when Feed Publishing Type is set to *Scheduled*, this field determines the languages in which the system publishes the query feed. Select from these options:

Current Language Only: - Select to publish the query feed in the language of the content owner.

All Available Languages: - Select to publish the query feed in all translated languages in your database.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Query Feed Security Select from these options:

Public Feed: Select to make the query feed available to all users. This option is available with all Feed Publishing types.

Note. The user context for processing the query is the default user on the ANONYMOUS node.

Query Security: Select to use Query security as the basis for the feed-level security. This option is available only with the *Real Time* Feed Publishing type.

When you define this type of query feed security, only the users who have rights to modify the query in Query Manager can find and view the query feed.

DistributionList/QuerySecurity: Select to use the Query distribution list in Report Manager as the basis for feed-level security. This option is available only with the *Scheduled* Feed Publishing type.

When you define this type of query feed security, only the users that are on the distribution list when the query feed is scheduled to run from the Schedule Query Request dialog box can find and view the feed.

See Chapter 4, "Creating and Using Feeds," Feed Security Options, page 35.

Query Fields

This grid displays the fields that are available to the query as defined in the query definition in Query Manager. The grid is display-only.

Feed Entry Content Mapping

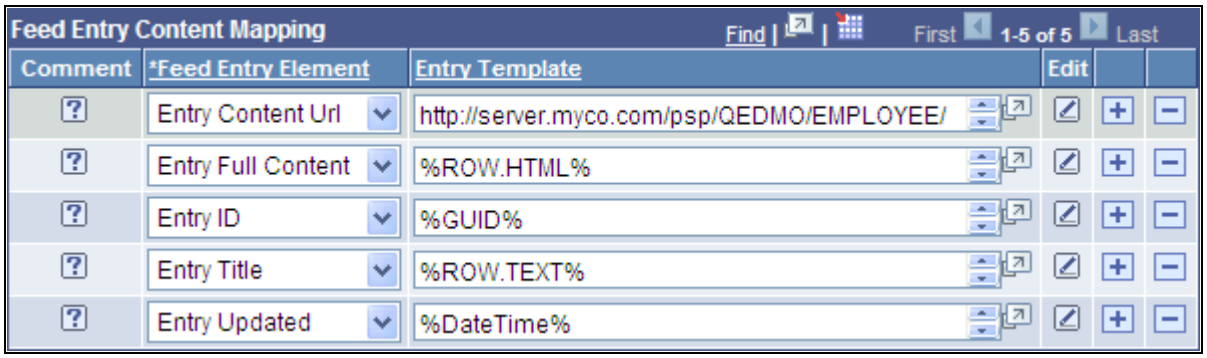
The fields in the Feed Entry Content Mapping grid on the Advanced Feed Options page enable you to create a template for the query data that is used to generate each feed entry. You select feed entry elements and either use the given template or build one of your own using the Feed Element Mapping Builder. This information, when published, determines how each entry appears to the user.

See Chapter 7, "Creating and Using Query Feeds," Mapping Feed Entry Elements to Entry Templates, page 70.

Mapping Feed Entry Elements to Entry Templates

The Feed Entry Content Mapping grid appears at the bottom of the PSQuery Data Type - Advanced Feed Options page.

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Feed Entry Content Mapping grid

You can use one of three methods to edit entry templates for feed entries:

- Directly in the Entry Template field in the Feed Entry Content Mapping grid.
- In a modal window when you click the Display Entry Element in Modal Window button.
- In Mapping Builder when you click the Edit button.

Important! Unless you are aware of the proper encodings to use, Oracle recommends that you use the Mapping Builder to edit or create entry templates.

See Chapter 7, "Creating and Using Query Feeds," Using Mapping Builder to Edit Entry Templates, page 75.

The Feed Entry Content Mapping grid has these fields:

Comment Click to display more information about this feed entry element in a modal window.

Feed Entry Element Each feed entry element is a property of an entry that can be interpreted and displayed by a feed reader.

Note. Not all feed readers display all elements. Consult the documentation on your feed reader.

Entry Template A template is a string that encloses bind variables inside % symbols. When the feed is published, the string as XML passes to the feed reader, which displays the content from the PSQuery data source according to the template instructions.

Entry templates support these bind variable types:

- *System variable,* for example, `%Copyright%`.
- *Template variable,* for example, `%ROW.HTML%`.
- *Query columns,* for example, `%ALIASNAME.QUERYCOLUMNNAME%`.
- *Message Catalog text,* for example, `%MSGT:10.10%`.
- *Message Catalog explanation,* for example, `%MSGE:10.10%`.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Display Entry Element in Modal Window

- On the grid itself, click to display the entire grid in a separate modal window.
- To the right of an entry template, click to display and edit the text of the entry template in a separate modal window.

Edit

Click to access the Feed Element Mapping Builder page to use the Mapping Builder to modify the entry template.

See Chapter 7, "Creating and Using Query Feeds," Using Mapping Builder to Edit Entry Templates, page 75.

This table describes the values that you can select for the Feed Entry Element field:

Feed Entry Element	*Description*	*Action of Feed Reader*	*Entry Template Values*	*Runtime Processing*
Entry Title	This element is a brief, single-line explanation of the feed entry. Only one entry title can exist per entry.	The reader displays the title as indicated by the template	Any template containing any bind variable. Can have HTML tags but needs to be only one line	The system escapes the HTML as well as XML characters.
Entry Content URL	This element is the URL to the PeopleSoft application page or other content. Only one entry content URL can exist per entry.	The reader displays an active link to enable the user to navigate to this URL.	Can map this to a field containing a URL, a computed field, or a static URL which has query columns as parameters	The system escapes no characters. Make sure that the URL is valid and confirm that you can use it as an attribute.
Entry Categories	This element defines the category of an entry. Multiple entries can belong to the same category and one entry can belong to multiple categories.	The reader organizes and filters entries based on values.	Any template containing any bind variable, but user should consider short values for ease of organization	The system escapes the HTML as well as XML characters.
Entry ID	This element is a permanent, universal identifier. Only one ID can exist per entry.	The reader detects changes to an existing entry and presents modifications of the existing entry instead of creating a new entry while ignoring unchanged entries.	Any template containing any bind variable	The system escapes the HTML as well as XML characters.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Entry Element	Description	Action of Feed Reader	Entry Template Values	Runtime Processing
Entry Updated	This element is the date and time that the entry was most recently modified. Only one update date can exist per entry.	The reader uses this entry to determine, in conjunction with the ID, whether to update the data in the entry.	Static value in PeopleTools date and time format. Dynamic value present in any date time query column. System variable `%DateTime`.	Values that are PeopleTools date time formatted will be converted to atom specific date time format.
Entry Published	This element is the date and time when the entry originally created. Only one publish date can exist per entry.	The reader uses this entry to determine the maximum age of the data.	Static value in PeopleTools date and time format. Dynamic value present in any date time query column. System variable `%DateTime`.	Values that are PeopleTools date time formatted will be converted to atom specific date time format.
Entry Author	This element pertains to information about the creator, owner, and author of this entry. Only one author can exist per entry.	The reader organizes and filters entries based on the value of this field. The reader can use this field to send email to author.	Static values in this format: `Name[Email⇒ Address]` Author name and email ID can be mapped to query columns or system variables.	The system extracts the email address extracts from [] and the name from the value before the brackets. **Note.** The brackets may be left empty, but must be present for the system to correctly process this entry. The system escapes the HTML as well as XML characters.
Entry Contributors	This element pertains to information about various parties involved in this entry. Multiple contributors can exist per entry.	The reader organizes and filters entries based on the value of this field. The reader can use this field to send email to author.	Static values in this format: `Name[Email⇒ Address]` Contributor name and email ID can be mapped to query columns or system variables.	The system extracts the email address extracts from [] and the name from the value before the brackets. **Note.** The brackets may be left empty, but must be present for the system to correctly process this entry. The system escapes the HTML as well as XML characters.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

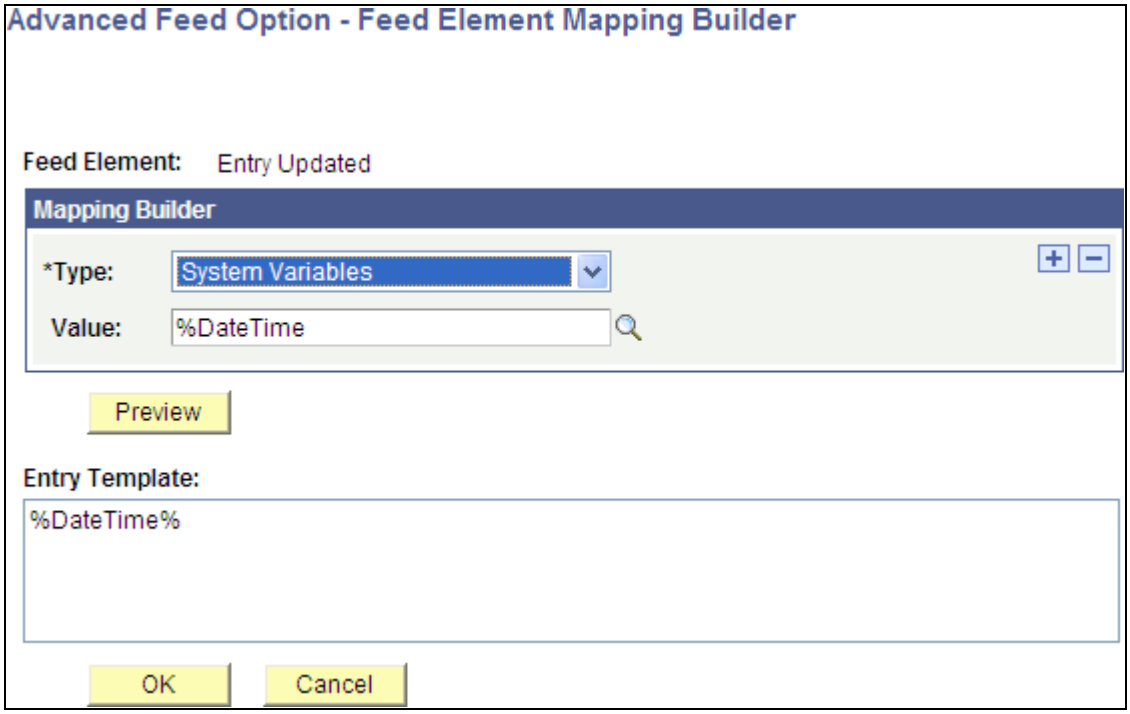
Feed Entry Element	Description	Action of Feed Reader	Entry Template Values	Runtime Processing
Entry Description	This element is a brief description of the about the entry, this can have more detailed information than a title and less information compared to full content	The reader presents more information than the title, but less than the full content while providing a link to the full content.	Any template containing any bind variable. HTML tags.	The system escapes the HTML as well as XML characters.
Entry Enclosures	This element contains other content types accessible using URL, for example images and mp3 files. Multiple enclosures signifying multiple attachments can exist per entry.	The reader can automatically download content and display it in place or enable the viewer to access it by using external tools based on the content type.	URL to the resource, its content, and its length in bytes. Use this format `""href="URL" type="content/type" length="byte-size""`. The system extracts the values inside the quotes. You can leave empty quotes in case the value needs to be ignored. You can map the href, type and length values that are within the quotes to any bind variable but the template output should be a valid URL/Content-type/length.	The system escapes the HTML as well as XML characters.
Entry Full Content	This element is the body of the entry. Only one full content can exist per entry.	The reader can display the content of the entry by default or upon user request.	Any template containing any bind variable. HTML tags.	The system escapes the HTML as well as XML characters.
Expires	This element is the date time when this entry is invalid and should be considered outdated. Only one expiration can exist per entry.	The reader determines when to discard this entry.	Static value in PeopleTools date and time format. Dynamic value present in any date time query column.	Values that are PeopleTools date time formatted will be converted to atom specific date time format. The system escapes the HTML as well as XML characters.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Entry Element	Description	Action of Feed Reader	Entry Template Values	Runtime Processing
Max Age	This element is the length of time in milliseconds until when the entry will be valid after the Published or Updated date time. Only one maximum age can exist per entry.	The reader determines when to discard this entry.	Static number or mapped to a query column which results in number.	The system escapes the HTML as well as XML characters.
Entry Copyright	This element contains the copyright information. Only one copyright can exist per entry.	The reader displays the copyright information upon request.	Any template containing any bind variable. The `%Copyright%` system variable can be used to map to the system-level copyright information.	The system escapes the HTML as well as XML characters.

Using Mapping Builder to Edit Entry Templates

Access the Feed Element Mapping Builder page (click the Edit button next to an entry template in the Feed Entry Content Mapping grid).



Feed Element Mapping Builder page

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Entry templates can be edited using Mapping Builder, which enables you to create custom output for feed entries using your own entry templates. You do this by assembling various combinations of static text and variable elements or by writing directly in the rich text editor.

Note. Currently, sharing templates among feed administrators is not supported. However, you can use the Download icon to export the grid values so that they can be used with other query feeds.

Assembling Entry Template Elements

These feed entry elements can be edited by assembling combinations of element types:

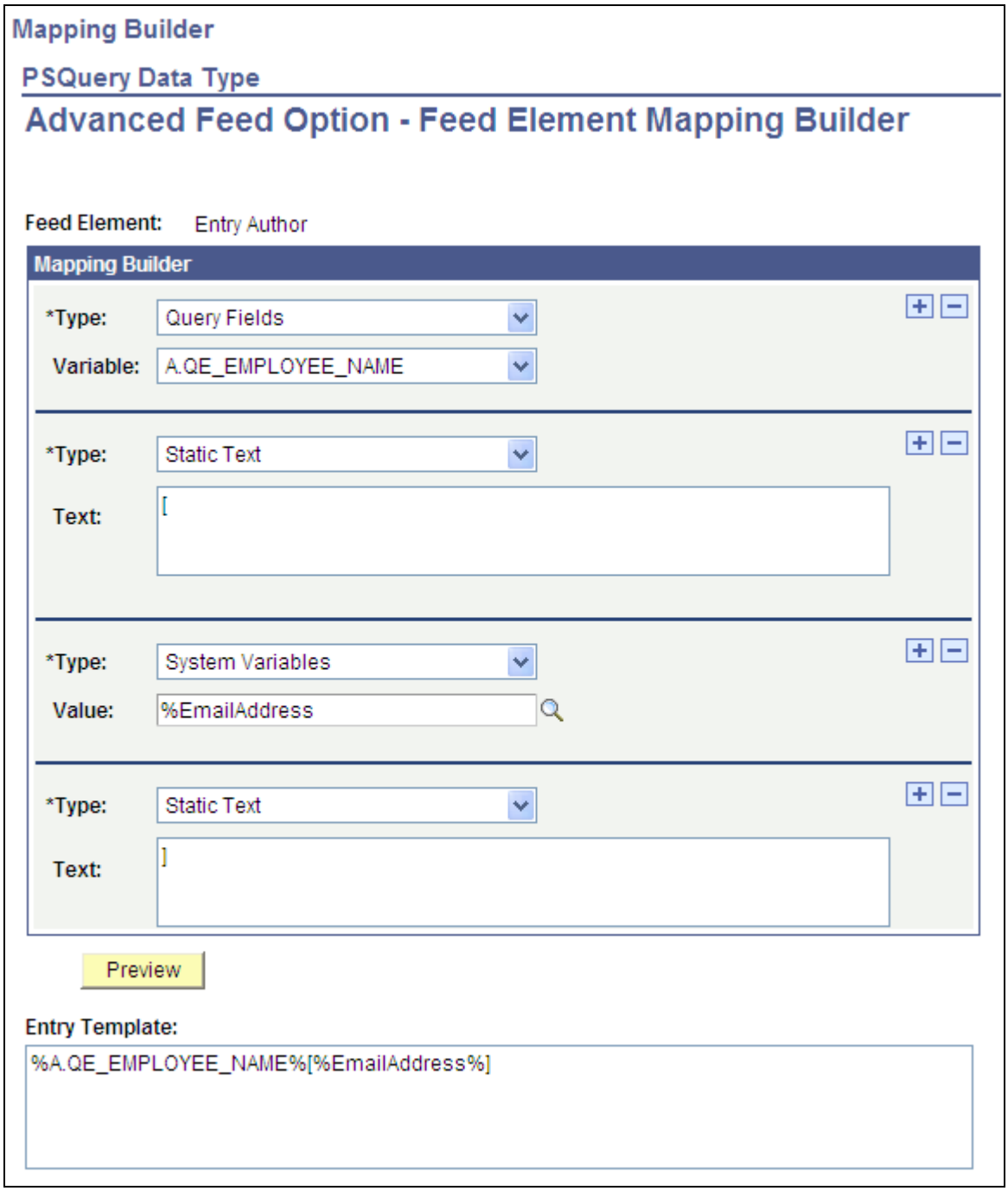
- Entry Author
- Entry Categories
- Entry Content URL
- Entry Contributors
- Entry Copyright
- Entry Enclosures
- Entry ID
- Entry Published
- Entry Updated
- Expires
- Max Age

To create an entry template:

1. On the Advanced Options page, click the Edit button associated with the entry template you want to edit.
2. In the Mapping Builder grid, select the Type of element to add to the entry template.
3. Click the + button to add another element.
4. Continue adding rows and constructing the entry template.
5. Click the Preview button.
6. Click the OK button.

This example shows an assembled entry template:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Example of the Feed Element Mapping Builder page showing an assembled entry template

The Mapping Builder fields are described as follows.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Type Select from these mapping builder types to assemble the entry template:

Message Catalog Explanation

Message Catalog Text

Query Fields

Static Text

System Variables

Templates

Message This field is visible when the Type field value is *Message Catalog Explanation* and *Message Catalog Text.*

Enter the message number to appear in the entry.

Note. Use the Message Catalog rather than static text to generate a template that is language independent.

Msg Set (message set) Visible when the Type field value is *Message Catalog Explanation* and *Message Catalog Text.*

Enter the message set number to appear in the entry.

Note. Use the Message Catalog rather than static text to generate a template that is language independent.

Text This field is visible when the Type field value is *Static Text.*

Enter the text that you want to appear in the entry.

Note. To have a percent sign appear in the text of the template, it needs to escaped using %%. For example, if the template reads %%, then at runtime the builder replaces it with %.

Value This field is visible when the Type field value is *System Variables.*

Select from any of the available system variables.

See *PeopleTools 8.51 PeopleBook: PeopleCode Language Reference*, "System Variables."

Variable This field is visible when the Type field value is *Templates* or *Query Fields.*

When the Type field value is *Templates,* select from these two Template options:

- *Row in HTML Format*
- *Row in TEXT Format*

When the Type field value is *Query Fields,* select from any field comprising the query definition.

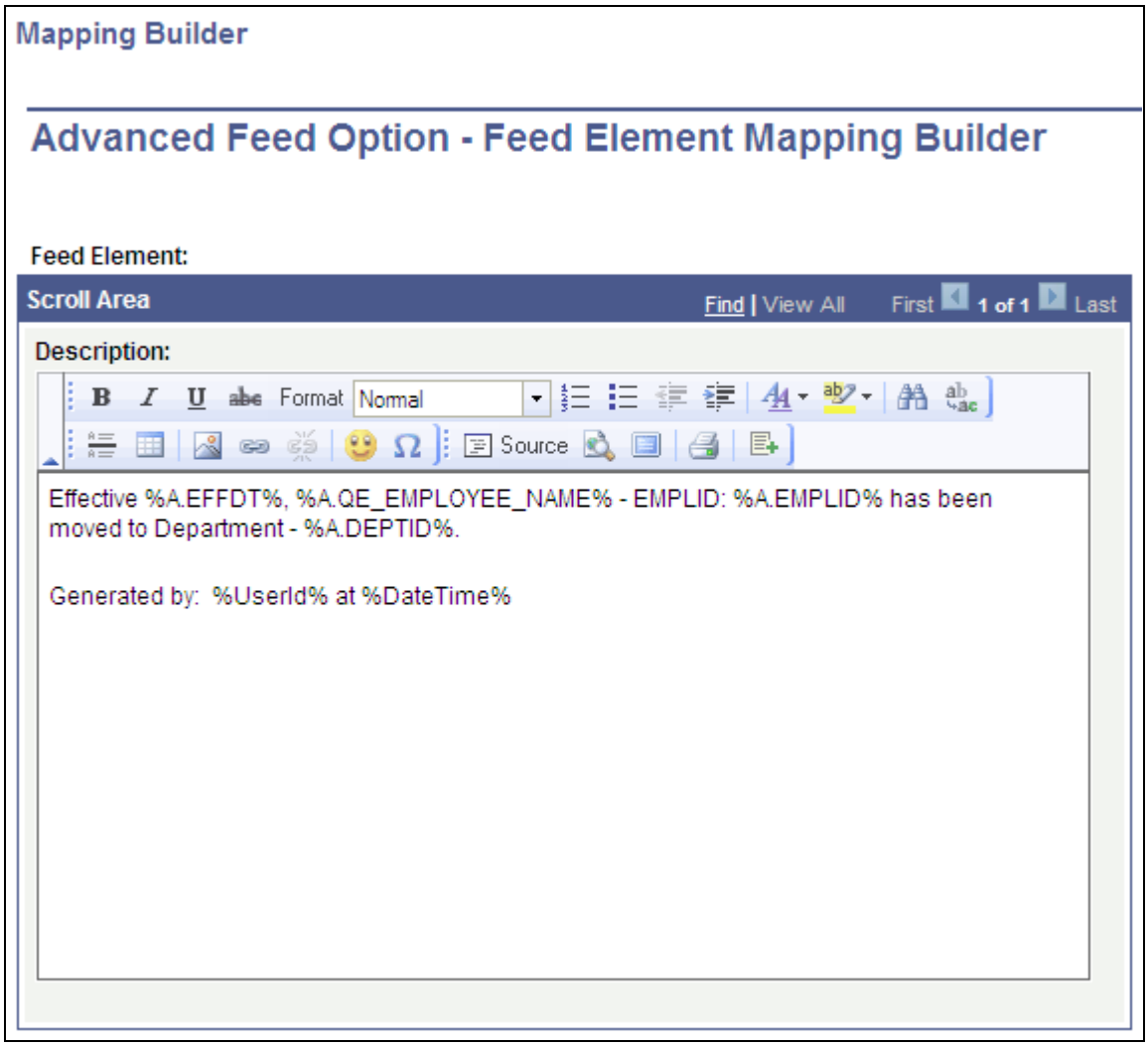
 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Creating Entry Templates in the Rich Text Editor

These feed entry elements present the rich text editor, where you can enter HTML, rich text, and bind variables to create a variety of output types to the feed reader:

- Entry Description
- Entry Full Content
- Entry Title

This example shows an entry template built by using the rich text editor:



Example of Feed Element Mapping Builder page showing an entry template built by using the rich text editor

Accessing Bind Variables in the Rich Text Editor

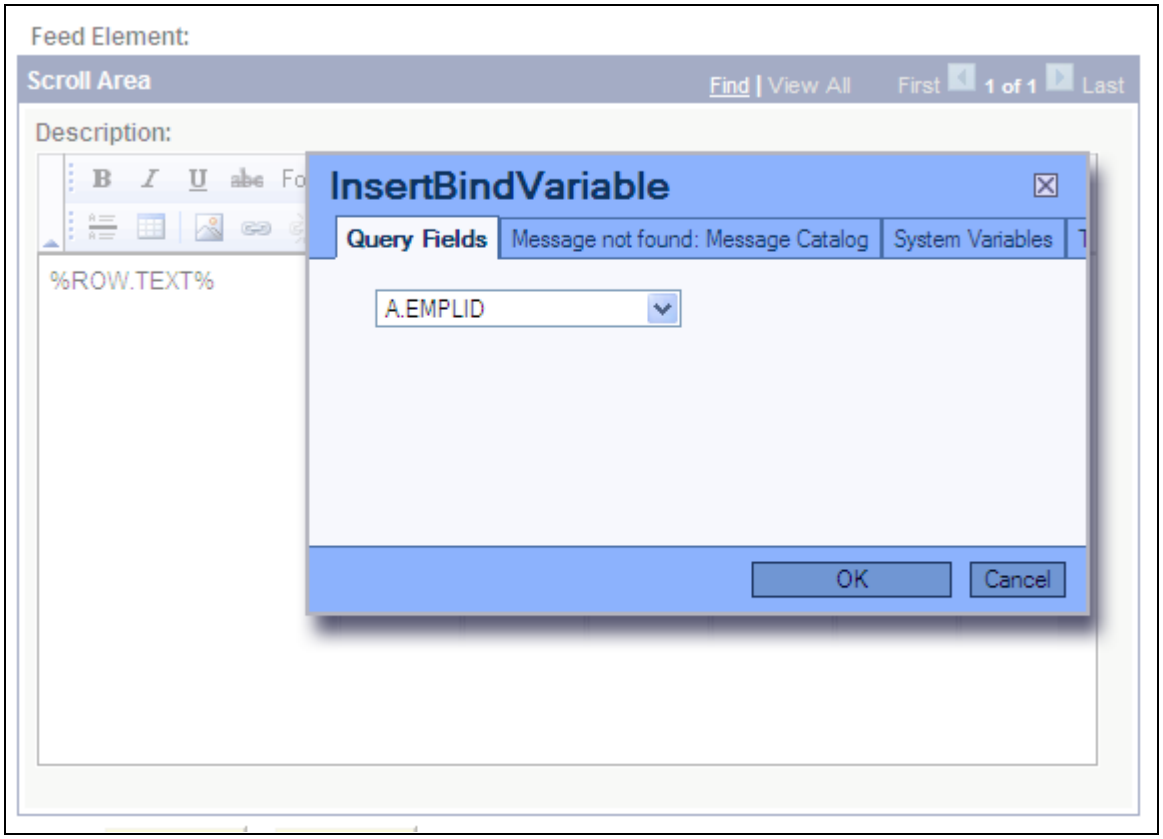
You can access these types of bind variables in the rich text editor:

- Query Fields

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Message Catalog
- System Variables
- Templates

Click the Insert Bind Variable button or right-click and select Insert Bind Variable to access the InsertBindVariable dialog box:



Example of InsertBindVariable dialog box

Using Query Feeds

You can view query feeds by selecting either of the following navigation paths:

- Access the My Feeds page.
  1. Select My Feeds in the Main Menu.
  2. Enter search criteria and click Search.
  3. Select the desired query feed.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Access Query Manager and use the related feeds hover menu.
    1. Select Reporting Tools, Query, Query Manager.
    2. Enter search criteria and click Search.
    3. Select a query that has already been published as a feed.
    4. From the query definition pages, select the feed from the related feeds hover menu.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 8

Creating and Using Worklist Feeds

This chapter provides an overview of worklist feeds and discusses how to:

- Publish worklist feeds
- Use worklist feeds

Understanding Worklist Feeds

PeopleTools provides the WORKLIST feed data type to publish worklist feeds through the Feed Publishing Framework.

Worklist feeds provide the ability to view a user's worklist as a feed in real time. The feed is generated in the security context of the signed on user and displays the user's current worklist items; however, it does not show worked items. By using worklist feeds, the worklist items can be available on systems other than the system that originated the entries without copying the data.

If worklist items are hosted on another system, then you must complete additional configuration in the local database for the node that represents that system.

Additional Configuration for Worklist Feeds

The local host node that contains the Worklist and Worklist Details pages must have defined content URI text and portal URI text. This configuration data will be used to generate the worklist item URLs. To find the node name, navigate to the Worklist page or the Worklist Details page and look at the URL, which will be similar to the following:

`http://machine:port/psp/ps/EMPLOYEE/ERP/w/WORKLIST?ICAction=ICViewWorklist&Menu=Worklist&Market=GBL&PanelGroupName=WORKLIST`.

In the previous URL, `ERP` is the local host node; therefore, you must define the content URI text and portal URI text for the `ERP` node so that worklist feeds contain valid feed entries.

See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Setting URI Text for Local Host Nodes, page 27.

Publishing Worklist Feeds

This section provides an overview of the steps used to publish a worklist feed and discusses how to:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Publish a worklist as a feed.
- Define advanced options for worklist feeds.

Understanding the Steps to Publish a Worklist Feed

To publish a worklist feed:

Note. Any user with the role *PeopleSoft Administrator* can publish worklist feeds from either the Worklist page or the Worklist Details page.

---

1. Select Worklist, Worklist to access the Worklist page.
2. Click the Publish as Feed link.

   See Chapter 8, "Creating and Using Worklist Feeds," Publishing a Worklist as a Feed, page 84.
3. On the Publish Feed Definition page, define the feed properties.

   See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.
4. Click the Advanced Options link.
5. Define the advanced feed options for the worklist feed.

   See Chapter 8, "Creating and Using Worklist Feeds," Defining Advanced Options for Worklist Feeds, page 85.
6. Click the OK button.
7. Click the Publish button.

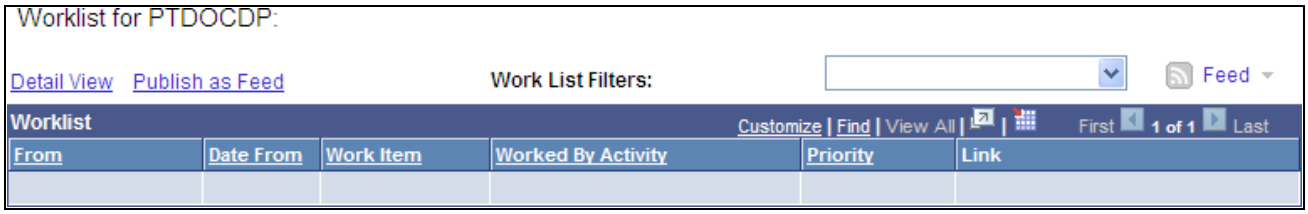
Pages Used to Publish Worklist Feeds

Page Name	*Definition Name*	*Navigation*	*Usage*
Worklist	WORKLIST	Worklist, Worklist	Create feed definitions for worklists.
Worklist	WORKLIST_DETAILS	Worklist, Worklist Details	Create feed definitions for worklists.
Worklist - Advanced Feed Options	WL_PUB_AS_ADVOPT	Click the Advanced Options link on the Publish Feed Definition page.	Define advanced options for worklist feeds.

Publishing a Worklist as a Feed

Access the Worklist page or the Worklist Details page (Worklist, Worklist; or Worklist, Worklist Details).

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Worklist page

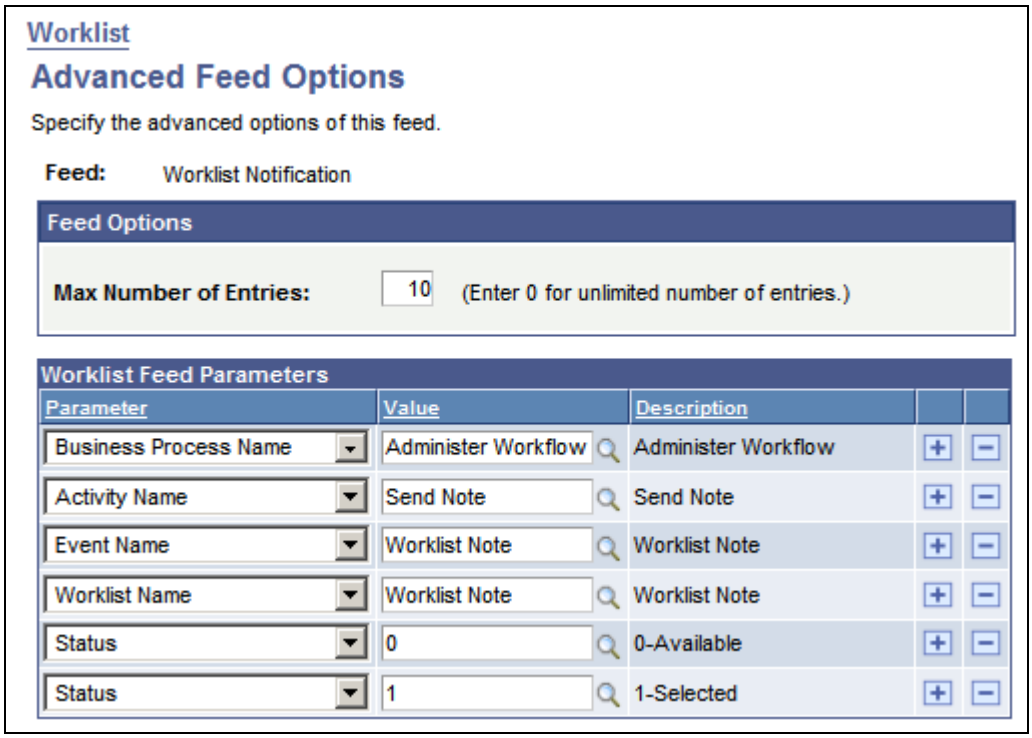
Use the Worklist page to publish a worklist as a feed. Click the Publish as Feed link to begin creating the feed definition.

Create the feed definition on the Publish Feed Definition page.

See Chapter 4, "Creating and Using Feeds," Defining Feed Properties, page 32.

Defining Advanced Options for Worklist Feeds

Access the Advanced Feed Options page (click the Advanced Options link on the Publish Feed Definition page).



Worklist - Advanced Feed Options page

Use the Advanced Feed Options page to determine which worklists and worklist items are to be published in the worklist feed.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Max Number of Entries Enter the maximum number of entries that the feed should return to the user. Enter *0* for unlimited entries up to the maximum row limit specified on the Feed Options page, which is 300 by default.

Note. The upper limit can be configured on the Feed Options page.

Worklist Feed Parameters Select the parameters and values for the worklists and worklist items that you want to publish. A status of *0–Available* or *1–Selected* will always be included in the feed parameters. After a worklist item is worked, it will no longer appear in the feed.

Note. Even if rows for statuses 0 and 1 are deleted from the grid, worklist feeds will always show all available and selected worklist items.

Reset to Defaults Click to reset any advanced options to their default values.

Worklist Feed Parameters

This table lists worklist feed parameters, selectable values, and their data source parameter names:

Worklist Parameter	*Values*	*Data Source Parameter*
Activity Name	Select from any workflow activity.	ACTIVITYNAME
Business Process Name	Select from any business process.	BUSPROCNAME
Event Name	Select from any event.	EVENTNAME
From	Select from any user.	ORIGINATORID
Priority	*1-High*,*2-Medium*, or *3-Low*	WL_PRIORITY
Status	*0-Available* or *1-Selected*	INSTSTATUS
Timed Out	*0-Available* or *1-Timed Out*	TIMEDOUT
Worklist Name	Select from any worklist.	WORKLISTNAME

Note. The Priority field denotes the priority that the user, that is the receiver of the notification, has assigned to the worklist item in the Worklist page and not the priority with which the worklist item was created.

When you specify multiple feed parameters, similar parameters use OR while distinct parameters use AND in the WHERE clause. For example, consider the following list of worklist feed parameters and runtime values:

Worklist Parameter	*Runtime Value*
Business Process Name	Administer Workflow

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Worklist Parameter	*Runtime Value*
Activity Name	Send Note
Event Name	Worklist Note
Worklist Name	Worklist Note
Priority	1
Status	1
Status	0
Timed Out	0
From	PSADMIN
From	PTSECADM

The WHERE clause for the SQL statement generated for the parameters shown in the preceding table is:

```
WHERE (BUSPROCNAME='Administer Workflow') AND (ACTIVITYNAME ='Send Note') AND  ⇒
(EVENTNAME='Worklist Note') AND (WORKLISTNAME='Worklist Note') AND ⇒
(WL_PRIORITY='1') AND ((INSTSTATUS=0) OR (INSTSTATUS=1)) AND (TIMEDOUT=0) AND⇒
((ORIGINATORID='PSADMIN') OR (ORIGINATORID='PTSECADM'))
```

This table describes how the Feed Publishing Framework automatically maps the feed properties to Atom 1.0 data source parameters:

Feed Property	*Data Source Parameter*
Author	ORIGINATORID and primary email address
Category	BUSPROCNAME, ACTIVITYNAME, EVENTNAME, WORKLISTNAME, WL_PRIORITY, TIMEDOUT, INSTSTATUS, ORIGINATORID
ContentURL	Worklist entry URL
Copyright	none
Contributor	none
Description	BUSPROCNAME | "/" | ACTIVTYNAME | "/" | EVENTNAME Created On: INSTAVAILABLEDTTM Instance ID: INSTANCEID From: ORIGINATORID
Enclosure	none
FullContent	none

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Property	*Data Source Parameter*
GUID	Worklist URL | " " | INSTAVAILABLEDTTM
Published	INSTAVAILABLEDTTM
Title	INSTANCEID | "." | BUSPROCNAME | "/" | ACTIVTYNAME | "/" | EVENTNAME
Updated	LASTUPDDTTM

Note. INSTAVAILABLEDTTM is the time at which the worklist item was created.

LASTUPDDTTM is the time at which the worklist item was last updated or selected.

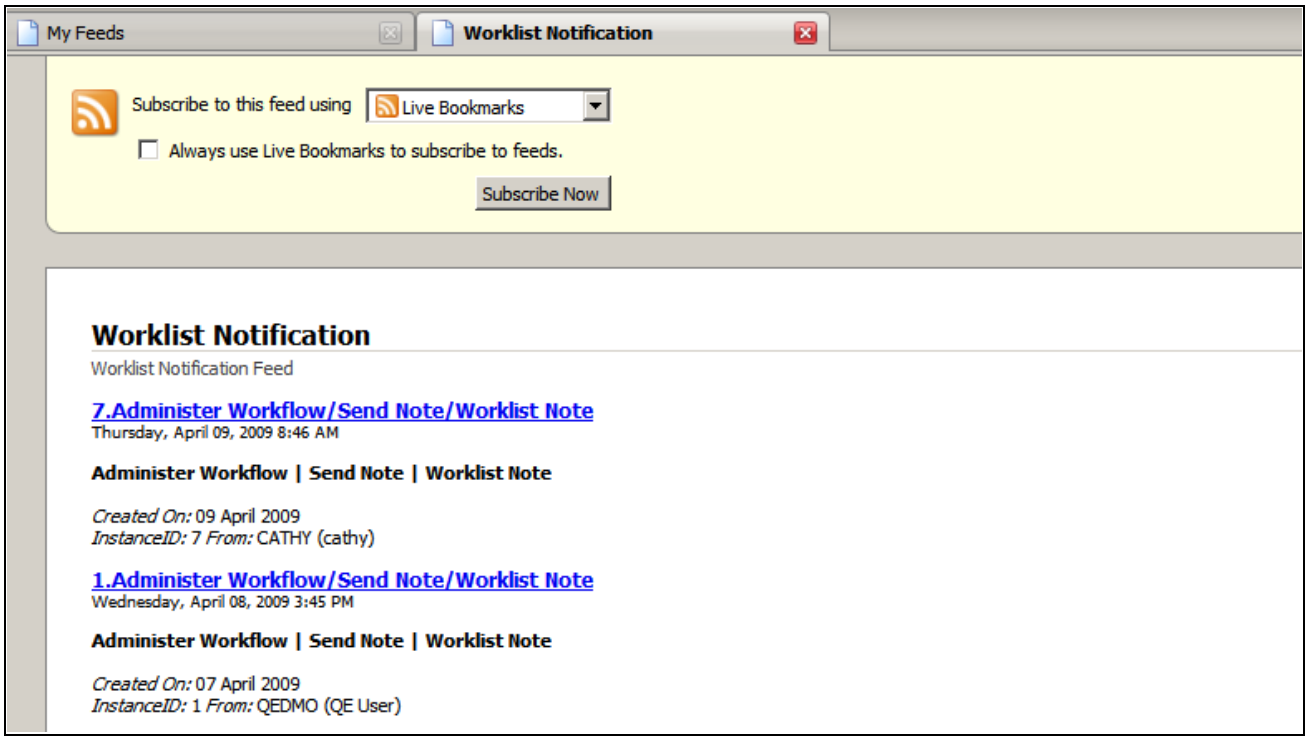
Using Worklist Feeds

You can view the feeds by selecting any of the following navigation paths:

- Access the My Feeds page:
  1. Select My Feeds in the Main Menu.
  2. Enter search criteria and click Search.
  3. Select the desired worklist feed.
- Access the Worklist page:
  1. Select Worklist in the universal navigation header.
  2. Select a worklist feed from the related feeds hover menu.
- Access the Worklist page:
  1. Select Worklist, Worklist.
  2. Select a worklist feed from the related feeds hover menu.
- Access the Worklist Details page:
  1. Select Worklist, Worklist Details.
  2. Select a worklist feed from the related feeds hover menu.

The following example shows a feed titled Worklist Notification in the browser:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Example of a worklist feed

When you click a worklist item, the appropriate page will appear.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Chapter 9

Developing New Feed Data Types

This chapter discusses:

- Analyze requirements for new feed data type.
- Create the feed data source application class.
- Define the feed data type.
- Update the property maintenance component.
- Update the view content component or pagelet.
- Examples of specific feed types.

Analyzing Requirements for New Feed Data Types

Consider these tasks when developing new feed data types:

- Determine how to distinguish feeds of the same data type (Data Source Settings). For example, the data source settings for Discussion Forums is the Forum ID, and for Content Management Folders the data source settings are the Portal Name and Folder ID.
- Determine how the feeds will be configured, and what the default value is for each parameter of the Data Source Parameters. For example, the data source parameter for Discussion Forums is Max Entries with a default of *10.* The data source parameters for Content Management Folders are Max Entries with a default of *10* and Include Subfolder Flags with a default of *yes.*
- Determine what to publish in the feed entry. For example, Discussion Forums publish the complete post, author, tags, attachment, and timestamp. Content Management Folders publish the content summary, author, tags, attachment, and timestamp.
- Determine whether the feed will be generated Scheduled or Real Time.
- Determine who can create and manage feed definitions.
- Determine where to put the Publish as Feed link.
- Determine where to put the related feeds hover menu.
- Determine how to handle the real-time feed security and the GETFEED viewer permission requests.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Creating the Feed Data Source Application Class

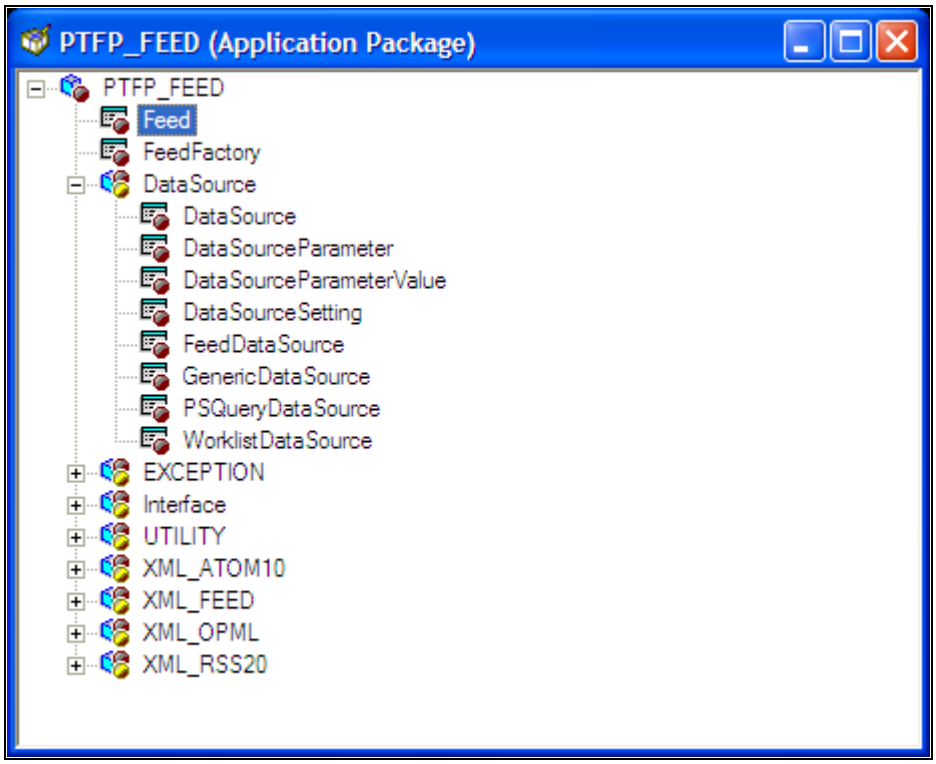
The feed data source application class manages all aspects of data collection and data security.

To create the feed data source application class:

- Extend the base class.
- Implement the methods.
- Set read only flags by using protected methods.

Extending the Base Class

Access and extend the PTFP_FEED:DataSource:DataSource base class.



PTFP_FEED application package showing DataSource base class

See *PeopleTools 8.51 PeopleBook: PeopleCode Developer's Guide*, "Creating Application Packages and Classes."

Implementing the Methods

Consider these method types:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Required
- Recommended
- Optional

Required Methods

This table describes the methods that you must implement:

Method	Purpose
`clone`	Clone the data source object
`getContentUrl`	Return the feed content url
`getDataSecurity`	Return the "allowed" list of viewer roles and/or permission lists
`isCurrentUserAuthorized`	Validate whether the current user has viewer permission of the feed
`initializeSettings`	Initialize data source setting collection and other class properties
`processSettingsChange`	Validate data source setting values, and generate data source parameter list accordingly
`execute`	Collect data based on user permission, and fill in the feed document

Recommended Methods

This table describes the methods that you should consider implementing:

Method	Purpose
`isCurrentUserAdmin`	Validate whether the current user has administration permission for the feed.
`copyProperties`	The clone method uses protected methods for copying class properties.

Optional Methods

This table describes the methods that you might consider implementing:

Method	Purpose
`onSave`	Perform tasks after saving the feed definition.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Method	*Purpose*
`onDelete`	Perform tasks before deleting the feed definition.
`getSettingDetail`	Return data source setting details as HTML.
`getParameterDetail`	Return data source parameter details as HTML.

Setting Flags by Using Protected Methods

You should set flags by using these methods:

- `setDataSourceType`
- `setSettingsCompleted`
- `setAllowRealTimeFeedSecurity`

Defining the Feed Data Type

This section lists the steps for defining new feed data types and discusses how to:

- Define a new feed data type.
- Determine whether there are additional advanced options.

Page Used to Define New Feed Data Types

Page Name	*Definition Name*	*Navigation*	*Usage*
Define Feed Data Types	PTFP_DATATYPE	PeopleTools, Feeds, Define Feed Data Types	Define feed data types.

Steps for Defining New Feed Data Types

You complete these steps to define a new feed data type:

1. Select PeopleTools, Feed, Define Data Types.
2. Specify the data source application class.
3. Select service operations to be used by feeds of this data type, and specify the default service operation.
4. Specify the default feed head level attributes for feeds of this data type.
5. (Optional) Click the Publish as Feed link to create a list of feeds feed, which lists all feeds of this data type accessible by the user.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

6. Determine whether there are additional advanced options.

Defining a New Feed Data Type

Access the Define Feed Data Types page (PeopleTools, Feeds, Define Feed Data Types).

Define Feed Data Types page

PeopleSoft applications contain four delivered feed data types: FEED, GENERICFEED, PSQUERY, and WORKLIST. To create a new feed data type, use Add New Value from the search page.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Data Type The type of feed that you are creating or editing. This field is display only.

Description Enter a short description of the type of feed. You may enter up to 30 characters.

Long Description Enter a long description of the type of feed to clearly clarify its purpose. You may enter up to 255 characters.

Active Select to activate the feed definition.

Default Feed Properties

Not all readers display all properties. This table describes the default feed properties that some feed readers process and display.

Note. These are default properties. You may change any of these default properties at the individual feed level.

Copyright Enter copyright information to be included in the XML.

Logo Enter a URL to the logo to be included in the XML, for example, *http://myserver.com/img/logo.gif*.

Icon Enter a URL to an icon to be included in the XML, for example, *http://myserver.com/img/icon.gif*.

Author Name Enter an author to be included in the XML.

Author Email Enter an author email address to be included in the XML.

Contributor Name Enter a contributor to be included in the XML.

Contributor Email Enter a contributor email address to be included in the XML.

Supporting Application Class

Package Name Enter the application class package name that you want to use for the data type.

Each Feed Data Type application class should be associated with one Feed Data Type service operation.

Path Enter the application class path that you want to use for the data type.

Application Class ID Enter the name of the application class that you want to use for the data type. The class must exist in the application package name that you specify.

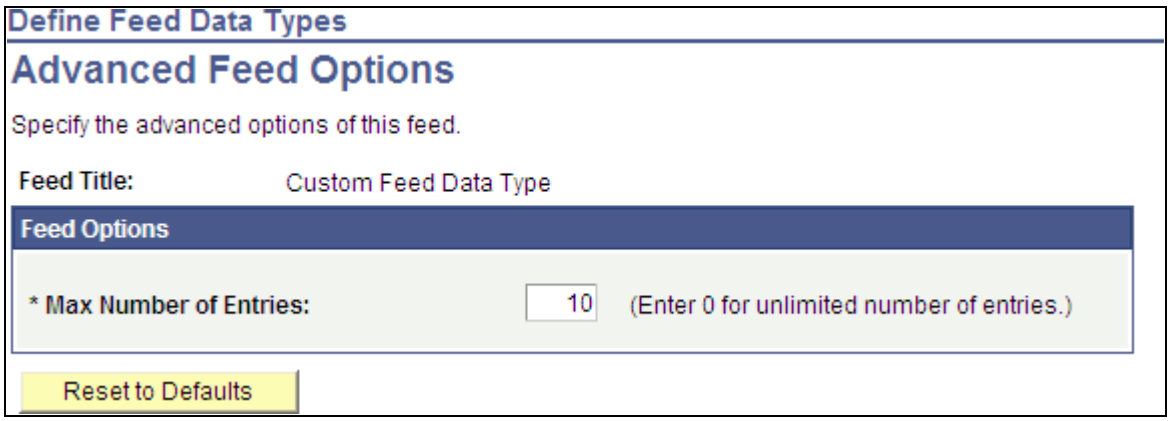
Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Feed Service Operations

Service Operation	Enter the name of the service operations associated with the feed definition that are used to retrieve data.
Type	This field displays whether the service operation is real time or scheduled. This field is display only.
Default	Select this check box to make this service operation the default.

Determine Whether There Are Additional Advanced Options

The standard advanced option page, PTFP_PUB_AS_ADVOPT, has one advanced option: Max Number of Entries. If your new feed data type has additional data source parameters (such as a paged feed, an incremental feed, or other parameters), then you must create a custom advanced options page. Otherwise, you can use the standard advanced options page shown in the following example:



Advanced Feed Options page (standard)

See Also

Chapter 9, "Developing New Feed Data Types," Creating an Advanced Options Page, page 100

Updating the Property Maintenance Component

You complete the tasks described below to update the property maintenance component.

This section discusses how to:

- Add the four standard Publish as Feed pages.
- Add the Publish as Feed link to one of the pages in the component.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

- Create an advanced options page.
- Add record PeopleCode.

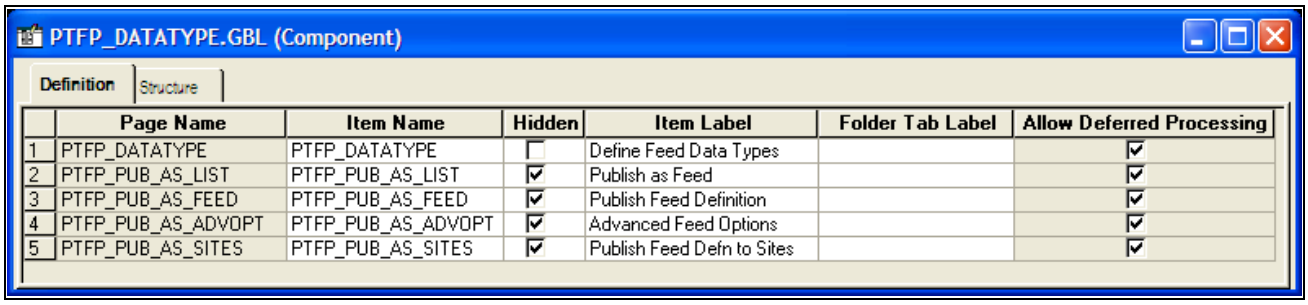
Adding the Four Standard Publish as Feed Pages

You must add these four standard, hidden Publish as Feed pages to the component:

- Publish Feed Definition (PTFP_PUB_AS_FEED)
- Advanced Feed Options (PTFP_PUB_AS_ADVOPT)
- Publish as Feed (PTFP_PUB_AS_LIST)
- Publish Feed Definition to Sites (PTFP_PUB_AS_SITES)

Note. All four pages can be cloned and then modified to suit unique requirements of the new feed data type.

Access the component, add the pages, and configure the pages as hidden as shown in this example:



	Page Name	Item Name	Hidden	Item Label	Folder Tab Label	Allow Deferred Processing
1	PTFP_DATATYPE	PTFP_DATATYPE	☐	Define Feed Data Types		☑
2	PTFP_PUB_AS_LIST	PTFP_PUB_AS_LIST	☑	Publish as Feed		☑
3	PTFP_PUB_AS_FEED	PTFP_PUB_AS_FEED	☑	Publish Feed Definition		☑
4	PTFP_PUB_AS_ADVOPT	PTFP_PUB_AS_ADVOPT	☑	Advanced Feed Options		☑
5	PTFP_PUB_AS_SITES	PTFP_PUB_AS_SITES	☑	Publish Feed Defn to Sites		☑

Example of PTFP_DATATYPE component showing four hidden standard Publish as Feed pages

See *PeopleTools 8.51 PeopleBook: PeopleSoft Application Designer Developer's Guide*, "Creating Component Definitions," Adding Pages to Components.

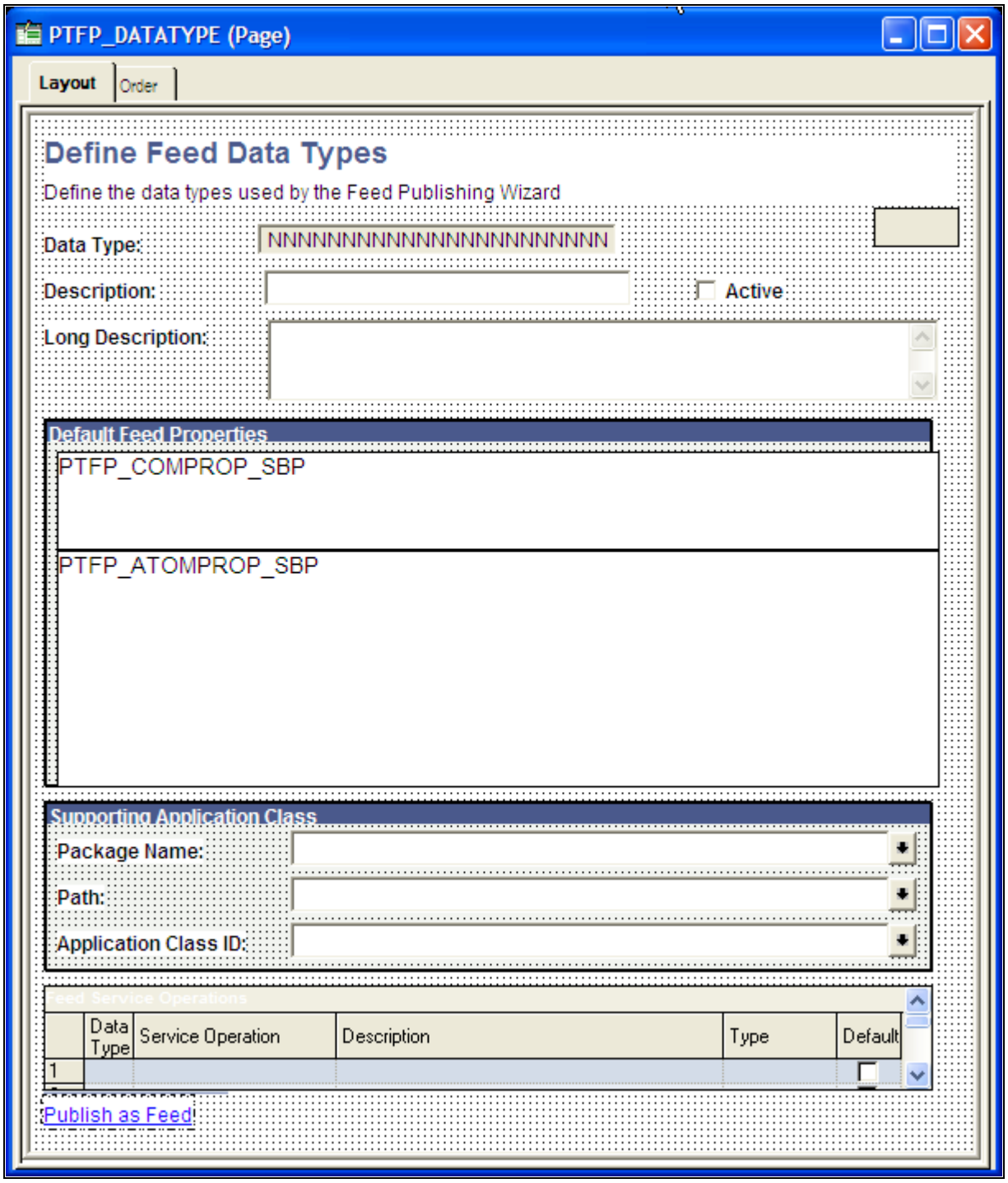
Adding the Publish as Feed Link to a Page

To publish the new feed data type, the feed administrator must have access to the Publish as Feed pages. To access these pages, you must add the Publish as Feed link to a page in the component.

Example

Notice the Publish As Feed link in the lower left corner of the PTFP_DATATYPE page:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



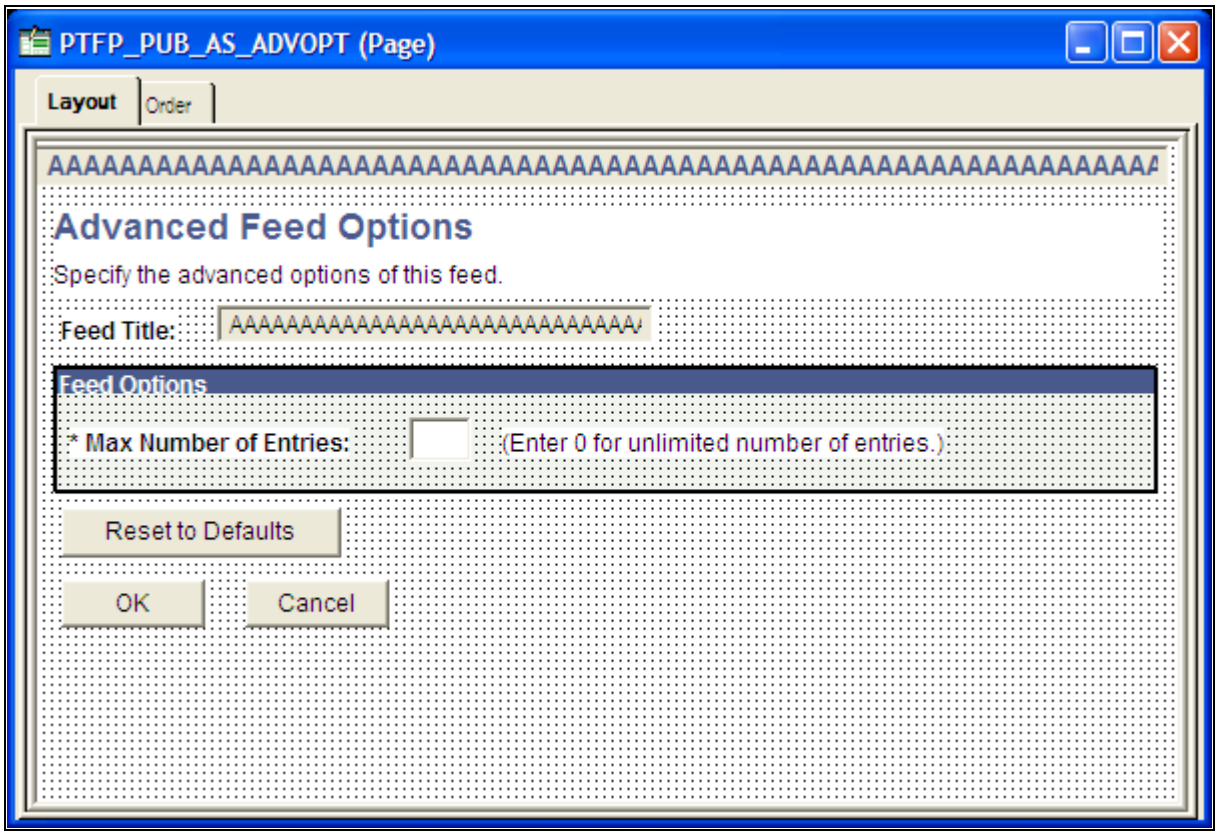
PTFP_DATATYPE page showing the Publish as Feed link

See *PeopleTools 8.51 PeopleBook: PeopleSoft Application Designer Developer's Guide*, "Using Page Controls," Using Push Buttons and Links.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Creating an Advanced Options Page

The advanced options page is used to set data source parameters for each feed definition. The standard advanced option page, PTFP_PUB_AS_ADVOPT, has one advanced option: Max Number of Entries. If your new feed data type has additional data source parameters (such as a paged feed, an incremental feed, or other parameters), then you must create a custom advanced options page. Otherwise, you can use the standard advanced options page shown in the following example:



PTFP_PUB_AS_ADVOPT page (the standard advanced options page)

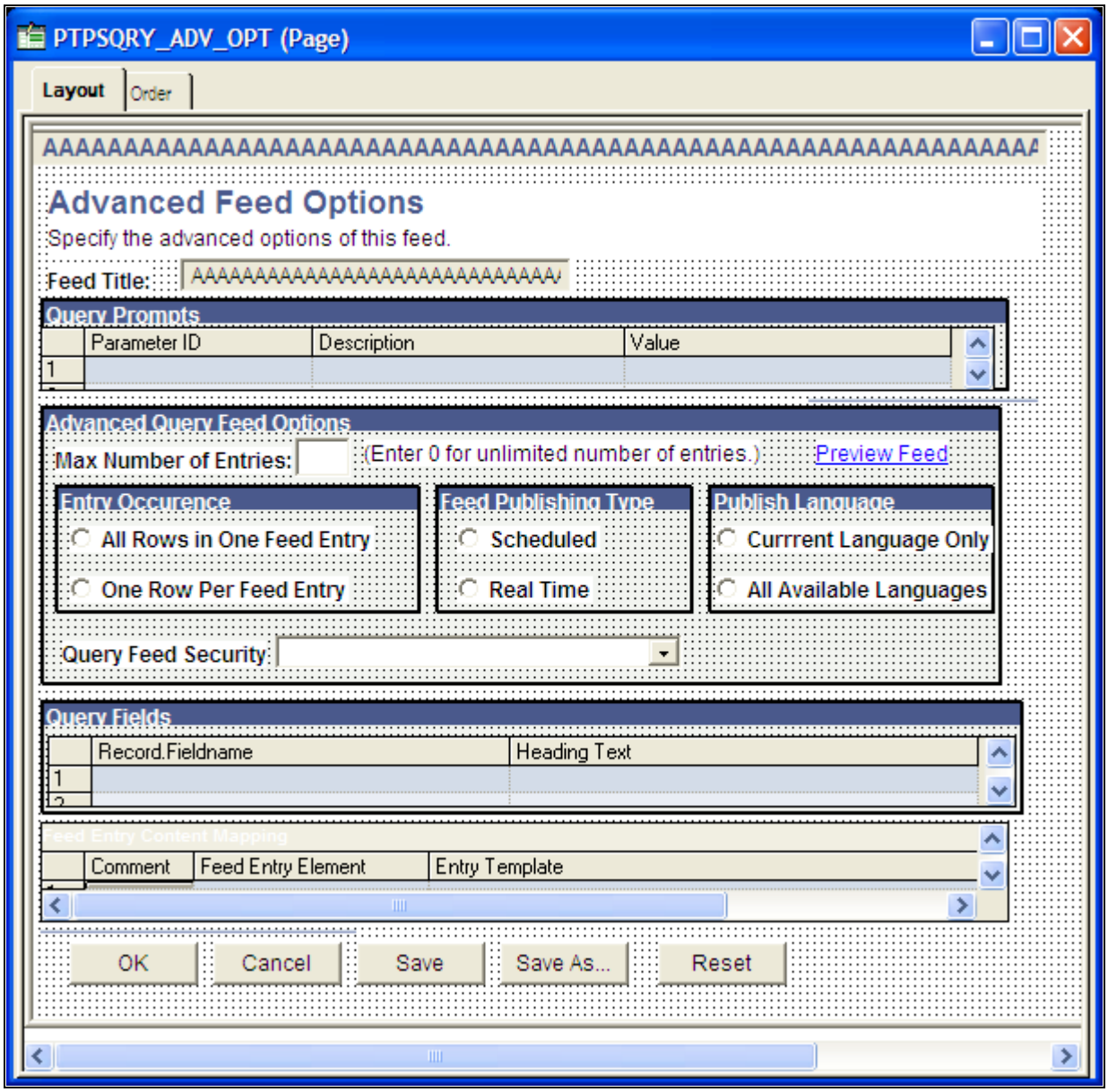
To create a custom advanced options page:

1. Clone the PTFP_PUB_AS_ADVOPT page as a feed data type-specific advanced options page.
2. Add the feed data type-specific data source parameters to the page.
3. In the page Activate event, create a PeopleCode program to read the data source parameter values from the feed definition.
4. Create an additional PeopleCode program to set the data source parameter values to the feed definition when the page is closed.

Example

This is the Query Advanced Options page. Notice how this page differs from the standard page.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



Query feeds advanced options page

See Also

PeopleTools 8.51 PeopleBook: PeopleSoft Application Designer Developer's Guide, "Creating Page Definitions"

Adding Record PeopleCode

Add the following code to the FieldChange event for the "Publish as Feed" component record field:

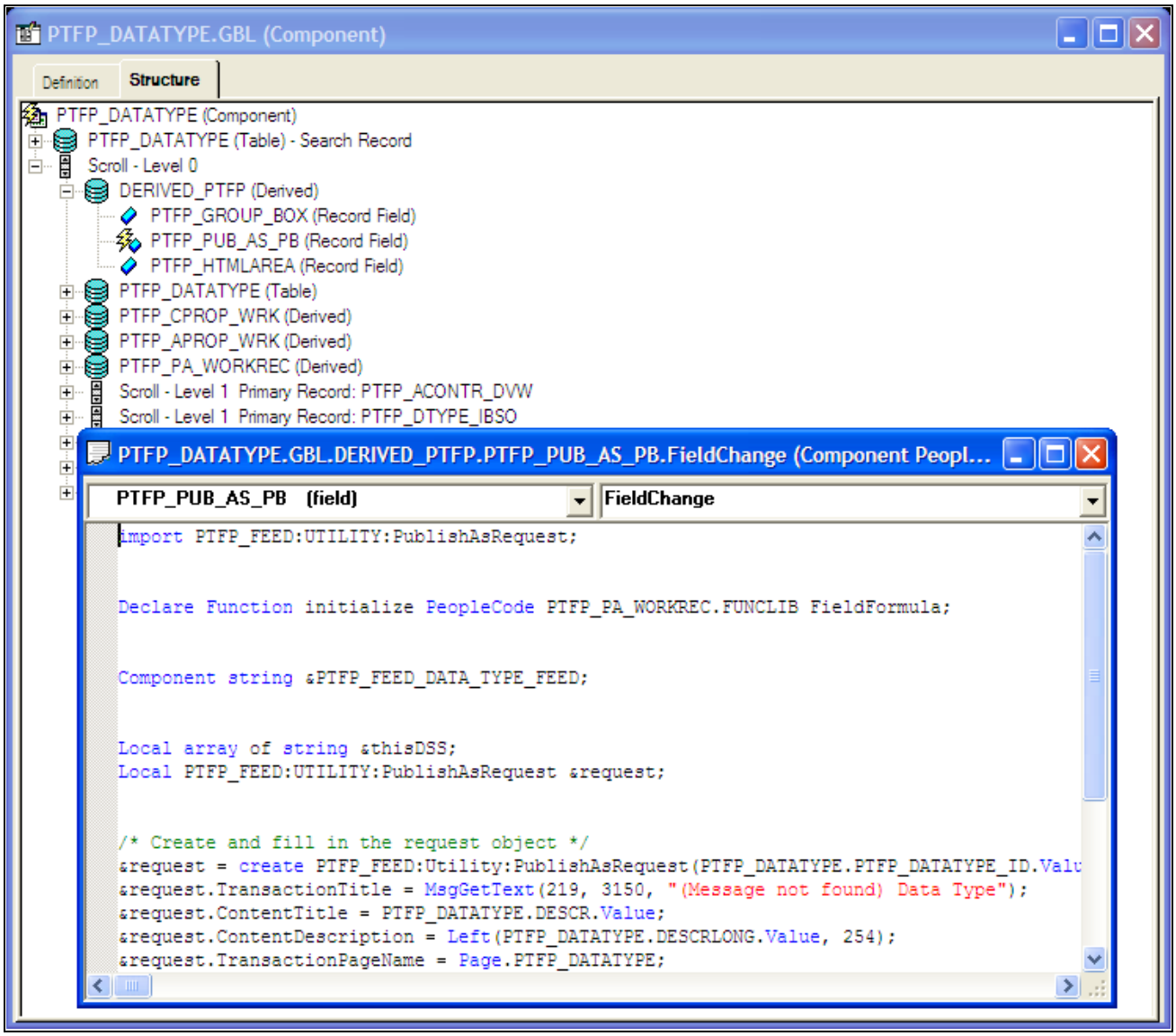
Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

```
import PTFP_FEED:UTILITY:PublishAsRequest;
Declare Function initialize PeopleCode PTFP_PA_WORKREC.FUNCLIB FieldFormula;
Local PTFP_FEED:UTILITY:PublishAsRequest &request;
Local array of string &thisDSS;

/* Create and fill in the request object */
&request = create PTFP_FEED:Utility:PublishAsRequest("<unique ID>");

&request.TransactionPageName = Page.<Page Name>;
&request.TransactionTitle = "<Page Title>";
&request.ContentTitle = "<Default Feed Title>";
&request.ContentDescription = "<Default Feed Description>";
&request.AdvancedOptionsPageName = Page.<Page Name>;
&request.DataTypeID = "yourDataTypeID";

/* Fill in the data source setting values */
&thisDSS = CreateArray("<Data Source Setting Name>", "<Data Source ⇒
Setting Value>");
&request.DataSourceSettings.Push(&thisDSS);

/* Start the process */
initialize(&request);
```

Example

Notice the Component Record Field PeopleCode on the PTFP_PUB_AS_PB field in this example:

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



PTFP_PUB_AS_PB field showing PeopleCode on the FieldChange event

---

Note. Where you add the record PeopleCode depends on which record field is used for the Publish as Feed page. If you use the standard page (DERIVED_PTFP.PTFP_PUB_AS_PB), you should only add code in the component record field FieldChange event. If the field is in your own work record, you can use the record FieldChange event directly

---

See Also

PeopleTools 8.51 PeopleBook: PeopleCode Developer's Guide, "Accessing PeopleCode and Events," Accessing Record Field PeopleCode

PeopleTools 8.51 PeopleBook: PeopleCode Developer's Guide, "Accessing PeopleCode and Events," Accessing Component Record Field PeopleCode

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Updating the View Content Component or Pagelet

This section discusses how to:

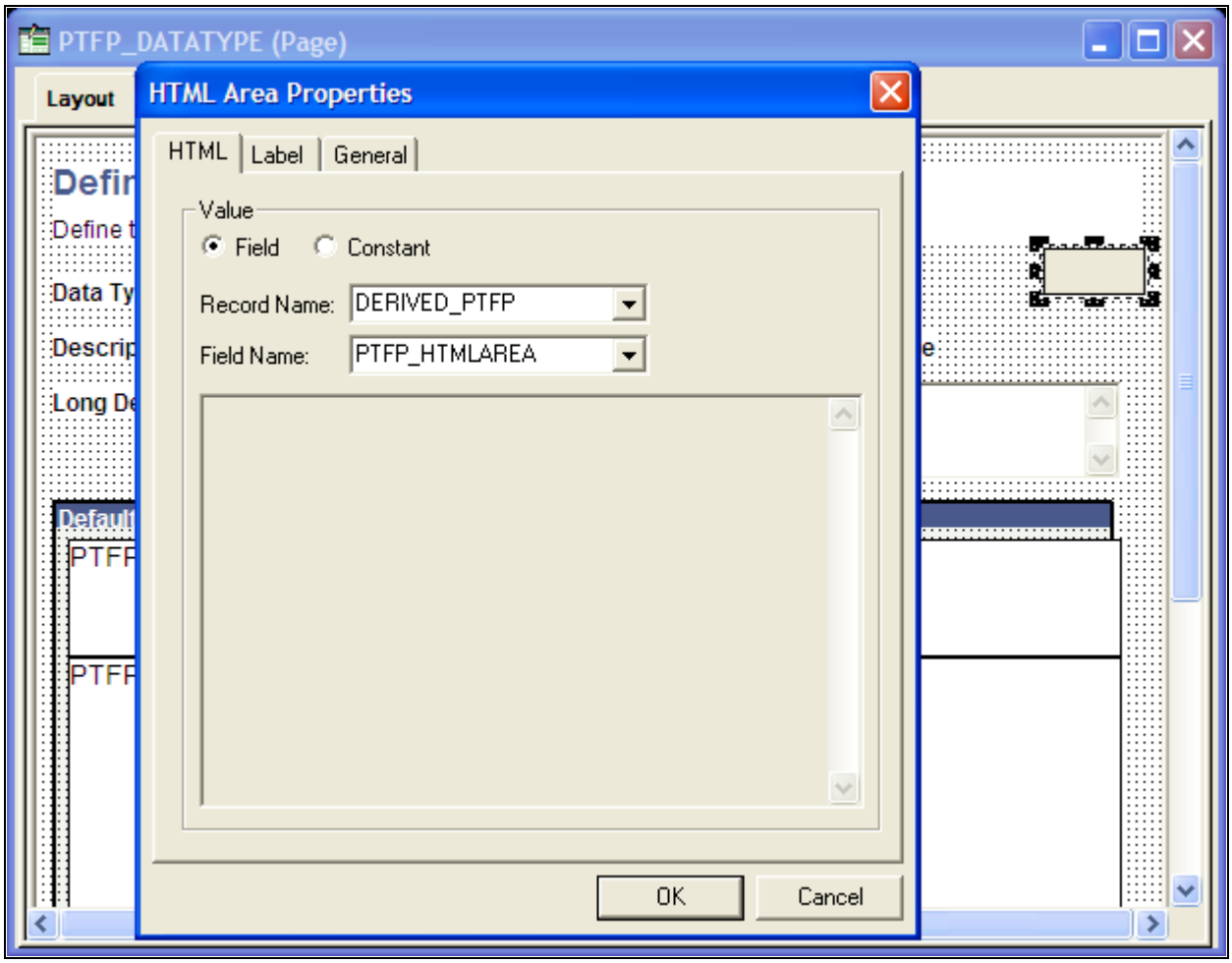
- Add the related feeds hover menu to pages.
- Add component or page PeopleCode.
- Add the related feeds hover menu to Pagelet Wizard pagelets.

Adding the Related Feeds Hover Menu to Pages

Add an HTML area to the page for hosting the related feeds hover menu.

Example

Notice the HTML area in the upper right of the PTFP_DATATYPE page:



PTFP_DATATYPE page showing HTML area and HTML area properties dialog box

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

See Also

PeopleTools 8.51 PeopleBook: PeopleSoft Application Designer Developer's Guide, "Using Page Controls," Using HTML Areas

Adding Component or Page PeopleCode

Add the following PeopleCode to the Activate event for the page:

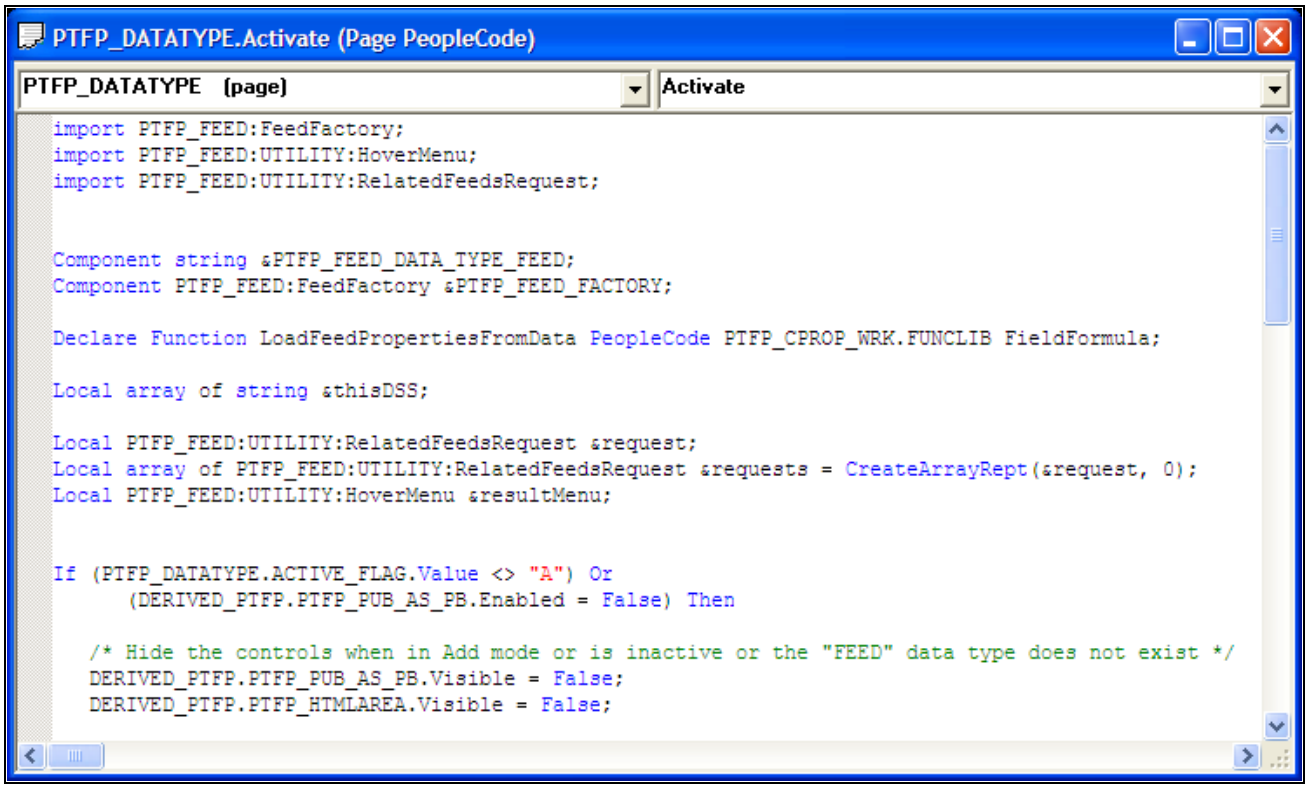
```
import PTFP_FEED:FeedFactory;
import PTFP_FEED:UTILITY:HoverMenu;
import PTFP_FEED:UTILITY:RelatedFeedsRequest;

Local PTFP_FEED:FeedFactory &PTFP_FEED_FACTORY;
Local PTFP_FEED:UTILITY:RelatedFeedsRequest &request;
Local array of PTFP_FEED:UTILITY:RelatedFeedsRequest &requests;
Local PTFP_FEED:UTILITY:HoverMenu &resultMenu;

&PTFP_FEED_FACTORY = create PTFP_FEED:FeedFactory();

/* Fill in the search criteria */
&request = create PTFP_FEED:UTILITY:RelatedFeedsRequest("<Unique ID>");
&request.DataTypeID = "<yourDataTypeID>";
&request.DataSourceSettings.Push(CreateArray("<Data Source Setting Name>", ⇒
"<Data Source Setting Value>"));
&requests.Push(&request);

try
   /* Generate the menu */
   &resultMenu = &PTFP_FEED_FACTORY.getRelatedFeedsHoverMenu(&requests);
   DERIVED_PTFP.PTFP_HTMLAREA.Value = &resultMenu.getHtml();
catch Exception &e
   WinMessage(&e.ToString(), %MsgStyle_OK);
end-try;
```

See *PeopleTools 8.51 PeopleBook: PeopleCode Developer's Guide*, "Accessing PeopleCode and Events," Accessing Page PeopleCode.

Example

In this example, you see the Page PeopleCode on the Activate event:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.



PTFP_DATATYPE Page PeopleCode example

Adding the Related Feeds Hover Menu to Pagelet Wizard Pagelets

When adding the hover menu, consider these points:

- Only homepage pagelets and embeddable pagelets on application pages support the related feeds hover menu.
- Transformer output must be XML or XHTML.
- Multi-group feed menu is supported.

Related Feeds Hover Menu XSL Example

This is sample XSL:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:variable name="NumRows">
  <xsl:value-of select="count(/queryresult/queryrows/row)" />
</xsl:variable>

<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="1" width="100%" summary="">
    <xsl:if test="$NumRows=0">
      <tr><td class="PSTEXT">no data available</td></tr>
    </xsl:if>

    <xsl:if test="$NumRows>0">
      <tr><td>
        <PSRELATEDFEEDSLINK>
          <feed id="ADMN_LIST_OF_FEEDS" />
          <feeds>
            <label>PSQUERY Feed List Feeds</label>
            <description>List of all PSQUERY feed list feeds</description>
            <feedDataType id="FEED" />
            <dataSourceSetting id="PTFP_DATATYPE_ID" value="PSQUERY" />
          </feeds>
        </PSRELATEDFEEDSLINK>
      </td></tr>
    </xsl:if>
  </table>
</xsl:template>

</xsl:stylesheet>
```

Examples of Specific Feed Types

This section provides examples of specific steps required when developing these feed types:

- Up-front scheduled feeds
- Real-time incremental feeds
- Paged feeds

Up-front Scheduled Feeds

Some additional steps are required for creating up-front scheduled feeds

You must complete the following tasks when creating up-front scheduled feeds:

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

1. The service operation that you use to publish the up-front feed messages to the Integration Broker queues must satisfy these conditions:

   - It should be an asynchronous, one-way service operation.
   - It should have PT_FEED_REQUEST.VERSION_1 as the message.
   - The service operation should be secured appropriately. This security is enforced by the scheduled feed GetFeed service operation handler at run time.
   - The queue used in the service operation should have the Archive option enabled, if the upfront feeds are to be archived. If this is *not* enabled, then the Archival Feeds will delete the feed messages in the Integration Broker queue.

2. The service operation used to publish the up-front feed messages has to be listed in the Define Feed Data Types page. This information is used for archiving feeds.

3. The feed format and language should be set as message attributes before publishing the message.

```
 /* Set the feed format of the message (i.e. Atom 1.0). */
&succeeded = &responseMsg.IBInfo.AddAttribute(&feedFactory.Utility.⇒
QUERYPARAMETER_FEEDFORMAT, &feedDoc.FeedFormat);

/* Set the language of the message. */
&succeeded = &responseMsg.IBInfo.AddAttribute(&feedFactory.Utility.⇒
QUERYPARAMETER_LANGUAGE, %Language);
```

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

4. DSPARAMETER_MAXROW, DSPARAMETER_SF_PAGING, DSPARAMETER_INCREMENTAL, and DSPARAMETER_SF_MAXMINUTES data source parameters found in the PTFP_FEED:UTILITY:Utility application class, should be defined and set to appropriate values in your implementation of the data source's processSettingsChange method. DSPARAMETER_MAXROW and DSPARAMETER_SF_MAXMINUTES are required for archiving feeds. DSPARAMETER_SF_PAGING is for paged feeds; DSPARAMETER_INCREMENTAL is for incremental feeds.

   For example:

```
&thisDSP = %This.addParameter(&utility.DSPARAMETER_MAXROW, ⇒
String(&utility.SF_MAXROWOPTION_LATESTMSG));
&thisDSP.Name = &thisDSP.ID;
&thisDSP.Description = MsgGetText(219, 3005, "Message Not Found - ⇒
Max Entries");
&thisDSP.FieldType = &utility.FIELDTYPE_NUMBER;
&thisDSP.DefaultValue = String(&utility.SF_MAXROWOPTION_LATESTMSG);
&thisDSP.Value = &thisDSP.DefaultValue;
&thisDSP.Required = True;

&thisDSP = %This.addParameter(&utility.DSPARAMETER_SF_PAGING, ⇒
String(&utility.SF_PAGINGOPTION_NOPAGING));
&thisDSP.Name = &thisDSP.ID;
&thisDSP.Description = MsgGetText(219, 3006, "Message Not Found - Paging");
&thisDSP.FieldType = &utility.FIELDTYPE_SIGNEDNUMBER;
&thisDSP.DefaultValue = String(&utility.SF_PAGINGOPTION_NOPAGING);
&thisDSP.Value = &thisDSP.DefaultValue;
&thisDSP.Required = True;

&thisDSP = %This.addParameter(&utility.DSPARAMETER_INCREMENTAL, ⇒
String(&utility.INCREMENTALOPTION_NO));
&thisDSP.Name = &thisDSP.ID;
&thisDSP.Description = MsgGetText(219, 3008, "Message Not Found - Incremental");
&thisDSP.FieldType = &utility.FIELDTYPE_SIGNEDNUMBER;
&thisDSP.DefaultValue = String(&utility.INCREMENTALOPTION_NO);
&thisDSP.Value = &thisDSP.DefaultValue;
&thisDSP.Required = True;

&thisDSP = %This.addParameter(&utility.DSPARAMETER_SF_MAXMINUTES, ⇒
String(&utility.SF_MAXMINUTES_ALLMSGS));
&thisDSP.Name = &thisDSP.ID;
&thisDSP.Description = MsgGetText(219, 3007, "Message Not Found - Max Min");
&thisDSP.FieldType = &utility.FIELDTYPE_SIGNEDNUMBER;
&thisDSP.DefaultValue = String(&utility.SF_MAXMINUTES_ALLMSGS);
&thisDSP.Value = &thisDSP.DefaultValue;
&thisDSP.Required = True;
```

   Important! The incremental feed option is incompatible with the paged feed option. Do not allow both options to be set simultaneously.

   ---

5. Modify the associated advanced feed options page to allow feed administrators the ability to set these options.

See Also

Chapter 5, "Administering Feeds," Archiving Feeds, page 55

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Real-Time Incremental Feeds

Some additional steps are required for creating real-time incremental feeds.

You must complete the following tasks when creating real-time incremental feeds:

1. The DSPARAMETER_INCREMENTAL data source parameter found in the PTFP_FEED:UTILITY:Utility application class must be defined and set to an appropriate value in your implementation of the data source's processSettingsChange method:

```
&thisDSP = %This.addParameter(&utility.DSPARAMETER_INCREMENTAL, ⇒
String(&utility.INCREMENTALOPTION_NO));
&thisDSP.Name = &thisDSP.ID;
&thisDSP.Description = MsgGetText(219, 3008, "Message Not Found - Incremental");
&thisDSP.FieldType = &utility.FIELDTYPE_SIGNEDNUMBER;
&thisDSP.DefaultValue = String(&utility.INCREMENTALOPTION_NO);
&thisDSP.Value = &thisDSP.DefaultValue;
&thisDSP.Required = True;
```

Important! The incremental feed option is incompatible with the paged feed option. Do not allow both options to be set simultaneously.

---

2. Modify the associated advanced feed options page to allow feed administrators the ability to set this option.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

3. Generate delta feed entries in your implementation of the data source's execute method based on the QUERYPARAMETER_IFMODIFIEDSINCE query parameter of the PTFP_FEED:UTILITY:Utility application class.

   Your implementation of the execute method must contain both the QUERYPARAMETER_IFNONEMATCH and the QUERYPARAMETER_IFMODIFIEDSINCE query parameters. QUERYPARAMETER_IFNONEMATCH is the feed ID and QUERYPARAMETER_IFMODIFIEDSINCE is the time at which the feed was last requested.

   The following code excerpt shows how to get the QUERYPARAMETER_IFNONEMATCH and QUERYPARAMETER_IFMODIFIEDSINCE query parameters using RequestInfo in the data source's execute method:

```
Local PTFP_FEED:DataSource:DataSourceParameter &thisDSP;
Local string &ifNoneMatch, &ifModifiedSince, &select;
Local datetime &lastmodified_dt = DateTime6(1900, 1, 1, 0, 0, 0);
Local boolean &incremental;

/* Get the Incremental Parameter */
&thisDSP = %This.getParameterById(&utility.DSPARAMETER_INCREMENTAL);
If &thisDSP <> Null And
(&thisDSP.EvaluatedValue = String(&utility.INCREMENTALOPTION_YES)) Then
   &incremental = True;
Else
   &incremental = False;
End-If;

&ifNoneMatch = &utility.RequestInfo.getParameter(&utility.QUERYPARAMETER_⇒
IFNONEMATCH);
&ifModifiedSince = &utility.RequestInfo.getParameter(&utility.QUERYPARAMETER_⇒
IFMODIFIEDSINCE);
If All(&ifModifiedSince) Then
   &lastmodified_dt = &utility.httpStringToDatetime(&ifModifiedSince);
End-If;

/* Compare and verify that &ifNoneMatch is same as the feed ID            */

/* Compare the &lastmodified_dt with appropriate datetime column like the */
/* LASTUPDDTTM field in the record used for generating the feed entries   */
```

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

4. When no feed entries are returned by a data source's execute method, the Feed Publishing Framework issues a 304-Not Modified HTTP header. If you are using custom feed handler—that is, a service operation different from the PTFP_GETFEED service operation —then use the setMessageHeadersAndMimeType method to set HTTP conditional headers.

   For example:

```
method OnRequest
   /+ &pRequestMsg as Message +/
   /+ Returns Message +/
   /+ Extends/implements PS_PT:Integration:IRequestHandler.OnRequest +/

   Local Message &responseMsg;
   Local XmlDoc &xmlDoc;
   Local string &temp, &errorText;

   Local PTFP_FEED:UTILITY:Utility &utility = &feedFactory_inst.Utility;
   Local PTFP_FEED:XML_FEED:FeedDoc &feedDoc;
   Local PTFP_FEED:UTILITY:FeedRequest &request;

   /* Ccreate the Search Request object */
   &request = create PTFP_FEED:UTILITY:FeedRequest("FeedRequest");

   ...

   try

      &feedDoc = &feedFactory_inst.getFeedDoc(&request);

   catch PTFP_FEED:EXCEPTION:NotFoundException &ex1
      &errorText = MsgGetExplainText(219, 3112, "(Message not found) Not Found");

   catch PTFP_FEED:EXCEPTION:PrivilegeException &ex2
      &errorText = MsgGetExplainText(219, 3113, "(Message not found) ⇒
Not Authorized");

   catch PTFP_FEED:EXCEPTION:FeedException &ex3
      &errorText = &utility.getExceptionText(&ex3);

   end-try;

   /* Create the response message */
   &responseMsg = CreateMessage(Operation.PTFP_GETFEED, %IntBroker_Response);

   If None(&errorText) Then
      &responseMsg = &utility.setMessageHeadersAndMimeType(&responseMsg, ⇒
&feedDoc, &request);
   Else
      &temp = "<?xml version='1.0' encoding='UTF-8'?><ErrorMessage>" | ⇒
&errorText | "</ErrorMessage>";
      &xmlDoc = CreateXmlDoc(&temp);
      &responseMsg.SetXmlDoc(&xmlDoc);
      &responseMsg.SegmentContentType = &utility.MIMETYPE_XML;
   End-If;

   Return &responseMsg;

end-method;
```

 Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Paged Feeds

Some additional steps are required for creating paged feeds.

Note. Paged feeds are supported for scheduled feeds only. The framework supports paged feeds via Integration Broker message segments. %MaxMessageSize is recommended when creating Integration Broker message segments for paged feeds.

You must complete the following tasks when creating paged feeds:

1. The DSPARAMETER_SF_PAGING data source parameter found in the PTFP_FEED:UTILITY:Utility application class must be defined and set to an appropriate value in your implementation of the data source's processSettingsChange method.

   For example:

   ```
   /* PAGING parameter */
   &thisDSP = %This.addParameter(&utility.DSPARAMETER_SF_PAGING, ⇒
   String(&utility.SF_PAGINGOPTION_NOPAGING));
   &thisDSP.Name = &thisDSP.ID;
   &thisDSP.Description = MsgGetText(219, 3007, "Message Not Found - Page Size");
   &thisDSP.FieldType = &utility.FIELDTYPE_SIGNEDNUMBER;
   &thisDSP.DefaultValue = String(&utility.SF_PAGINGOPTION_NOPAGING);
   &thisDSP.Value = &thisDSP.DefaultValue;
   &thisDSP.Required = True;
   ```

   Important! The paged feed option is incompatible with the incremental feed option. Do not allow both options to be set simultaneously.

   ---

2. Modify the associated advanced feed options page to allow feed administrators the ability to set this option.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Appendix A

Troubleshooting Tips

This appendix provides troubleshooting tips.

Common Problems

This table lists some common errors that may occur when setting up the feed framework:

Problem	*Resolution*
When you click a Publish as Feed link, the following pop-up error message is displayed: `Some required configuration steps are⇒ not done for the Feed Publishing⇒ Framework. Please check the IB Service⇒ Configuration.`	Complete configuration of your PeopleSoft system to support feeds. See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring Integration Broker Service Target Locations, page 23.
When you click the link for a feed—for example, on the My Feeds page—a new browser window opens and redisplays the page that you are on instead of the feed.	The feed was published even though configuration of the PeopleSoft system was incomplete. Complete configuration of your PeopleSoft system to support feeds. See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring Integration Broker Service Target Locations, page 23.
During configuration of Integration Broker, you get the following error message when you attempt to ping the default local node: `Integration Broker Authentication: No⇒ node Authentication Option set for⇒ node` *`node_name`* `on Target sys`	Set the node authentication to either password or certificate. See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring the Default Local Node, page 25.
When you click the link for a feed, one of the following error messages is displayed: • **`Address not found`** • **`Could Not Connect to Server`** • **`Unable to connect`** • **`Cannot display the webpage`**	The Target Location or the Secure Target Location is not set *properly* on the Service Configuration page. See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring Integration Broker Service Target Locations, page 23.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Problem	*Resolution*

When you click a feed entry in a feed, the following error message is displayed:
 	
`<?xml version="1.0" ?>`
`- <IBResponse type="error">`
`<DefaultTitle>Integration Broker⇒`
`Response</DefaultTitle>`
`<StatusCode>20</StatusCode>`
`<MessageSet>158</MessageSet>`
`<MessageID>505</MessageID>`
`<DefaultMessage>Unable to find a⇒`
`Routing corresponding to the incoming⇒`
`request message.</DefaultMessage>`
`<MessageParameters />`
`</IBResponse>`
	The Portal Context URI text or Portal URI text for the local host node has not been configured on the Node Definitions - Portal page.
	
	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Setting URI Text for Local Host Nodes, page 27.

When you open a worklist feed, feed entries are displayed there are no hyperlinks to view the details for each feed entry.	The Portal Context URI text or Portal URI text for the local host node for Workflow has not been configured on the Node Definitions - Portal page. See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Setting URI Text for Local Host Nodes, page 27. See Chapter 8, "Creating and Using Worklist Feeds," Additional Configuration for Worklist Feeds, page 83.

When you click a feed entry in a feed, the following error message is displayed:
 	
`Site name is not valid. Check your⇒`
`syntax and try again.`
	The Portal Context URI text or Portal URI text for the local host node is not configured *properly* on the Node Definitions - Portal page.
	
	Note. Both values require a terminating / to be correct.
	
	See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Setting URI Text for Local Host Nodes, page 27.

The feed document displays the feed title but no feed entries.	Feed entries appear in the context of the signed on user. One of the following could result in no feed entries in the feed document: • The current user has no feed entries—for example, the user does not have any unworked items in the specified worklist. • The feed definition was not specified correctly.
When you click the link for a feed, a new browser window is opened, but you are prompted to authenticate yourself with a user name and password: `A username and password are being⇒` `requested by` *`server_name`*	There is a mismatch in the way the system was identified in the signon URL versus how the system was identified for the Integration Broker service configuration. For example, this problem can occur when the signon URL is specified as a numeric IP address (that is, 10.123.123.789) and the service configuration is specified by a fully qualified domain name (that is, myserver.myco.com), or vice versa.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Problem	*Resolution*
When you click the link for a feed, a new browser window is opened, but you are prompted to authenticate yourself to the Oracle WebLogic Server with the following message: `The server server_name at WebLogic⇒ Server requires a username and⇒ password.`	Oracle WebLogic Server must be configured to disable its own authentication. See Appendix B, "Disabling Authentication on Oracle WebLogic Server," page 119.
When you request a feed, you are challenged for authentication even though you are already signed into the PeopleSoft system.	The PeopleSoft signon URL and the feed URL are of different protocols—for example, one uses HTTP and the other uses HTTPS. If a secure target location was specified on Integration Broker's Service Configuration page, then the PeopleSoft signon URL should be HTTPS. Conversely, if no secure target location was specified on the Service Configuration page and only a target location was specified, then the PeopleSoft signon URL should be HTTP. In either case, when a mismatch of protocols is used on your PeopleSoft system, you will be prompted for credentials again when you attempt to view a feed from a PeopleSoft page. **Note.** Certain integrations indicate the use of HTTPS. PeopleSoft feeds support integration with third-party feed readers through basic authentication only. However, since basic authentication isn't secure, HTTPS is recommended for feeds in this scenario. This can be set by specifying a secure target location on Integration Broker's Service Configuration page. See Chapter 3, "Configuring Your PeopleSoft System to Support Feeds," Configuring Integration Broker Service Target Locations, page 23.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Appendix B

Disabling Authentication on Oracle WebLogic Server

The Feed Publishing Framework requires that Oracle WebLogic Server's own authentication be disabled. Disabling Oracle WebLogic Server's authentication allows authentication to be passed through and handled by the PeopleSoft servlet. The config.xml file is shipped with this authentication disabled and therefore no action is required unless you have changed the setting in this file.

Note. IBM WebSphere does not require any special configuration steps.

Configuring Oracle WebLogic Server to Disable Authentication

By default, the delivered config.xml file is set to disable Oracle WebLogic Server's own authentication. No additional configuration is required unless you have changed this authentication setting.

To configure Oracle WebLogic Server to disable authentication, do the following:

1. Stop the web server.
2. Go to the *PS_HOME*\webserv*web_server*\config folder.
3. Edit the config.xml file.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

4. Add the following tag before the closing `</security-configuration>` tag:

   ```
   <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
   ```

   The edited file appears similar to the following with the added line highlighted below:

```
  <security-configuration xmlns:xacml="http://www.bea.com/ns/weblogic/90/security⇒
/xacml">
    <name>peoplesoft1</name>
    <realm>
      <sec:authentication-provider xsi:type="wls:default-authenticatorType"/>
      <sec:authentication-provider xsi:type="wls:default-identity-asserterType">
        <sec:active-type>AuthenticatedUser</sec:active-type>
      </sec:authentication-provider>
      <sec:role-mapper xsi:type="xacml:xacml-role-mapperType"/>
      <sec:authorizer xsi:type="xacml:xacml-authorizerType"/>
      <sec:adjudicator xsi:type="wls:default-adjudicatorType"/>
      <sec:credential-mapper xsi:type="wls:default-credential-mapperType"/>
      <sec:cert-path-provider xsi:type="wls:web-logic-cert-path-providerType"/>
      <sec:cert-path-builder>WebLogicCertPathProvider</sec:cert-path-builder>
      <sec:name>myrealm</sec:name>
    </realm>
    <default-realm>myrealm</default-realm>
    <credential-encrypted>{3DES}XLLC9Wru5qKeMAlvEULruO9LodVs7o3du4WVMtMs/⇒
ffmyP16aD4NKEv0va5IxytcWvGRV50mB5dYbzhos9XWNN0Lz4mQoXiy</credential-encrypted>
    <node-manager-username>system</node-manager-username>
    <node-manager-password-encrypted>{3DES}c/WblCLbZubUdNamvjN1sw==</node-⇒
manager-password-encrypted>
    <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-⇒
auth-credentials>
</security-configuration>
```

5. Save the file.

6. Restart the web server.

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

Index

A

B

C

D

E

F

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

G

H

I

L

M

N

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

O

P

Q

R

S

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.

T

U

V

W

Copyright © 1988, 2010, Oracle and/or its affiliates. All Rights Reserved.