

---

# Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework

---

**September 2009**

Copyright © 1988, 2009, 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

## Preface

<b>Oracle's PeopleSoft Feed Publishing Framework Preface .....</b>	<b>vii</b>
PeopleTools Products .....	vii
Feed Publishing Framework .....	vii
Integration Broker .....	vii
Query Manager .....	vii
Workflow .....	viii
Process Scheduler .....	viii

## Chapter 1

<b>Getting Started With PeopleSoft Feed Publishing Framework .....</b>	<b>1</b>
PeopleSoft Enterprise Feed Publishing Framework Overview .....	1
PeopleSoft Feed Publishing Framework Implementation .....	2
Prerequisites .....	2
Using PeopleTools-Delivered Feed Data Types .....	3
Developing New Feed Types .....	4

## Chapter 2

<b>Understanding the Feed Publishing Framework .....</b>	<b>5</b>
Feed Publishing Framework .....	5
Feed Publishing Runtime Engine .....	6
Feed Document Generation and Delivery .....	7
Feed Document Properties .....	8
Feed Data Type Application Classes .....	9
Creating a New Feed Data Type .....	10
Feed Data Types .....	11
List of Feeds (FEED) .....	11
Integration Broker Generic Message Feeds (GENERICFEED) .....	11
PS Query Feeds (PSQUERY) .....	11
Worklist Feeds (WORKLIST) .....	11
Real-Time and Scheduled Feeds .....	12
Real-Time Feeds .....	12
Scheduled Feeds .....	12

Feed Security .....	12
Security for Creating Feed Data Types .....	13
Security for Publishing Feeds .....	13
Security for Viewing Feeds .....	13
Feed Publication .....	14
My Feeds .....	14

## Chapter 3

<b>Creating and Using Feeds .....</b>	<b>17</b>
Defining and Publishing Feeds .....	17
Pages Used to Publish Feeds .....	17
Feed Publishing Business Process Flow .....	18
Publishing Feeds .....	19
Defining Feed Properties .....	19
Defining Advanced Options .....	22
Publishing Feeds to Additional Sites .....	23
Publishing a List of Feeds Feed .....	24
Accessing Feeds .....	25
Using the My Feeds Page .....	25
Using Related Feeds Hover Menus .....	27
Using the Get Feed List Service Operation .....	28
Viewing Feeds .....	30

## Chapter 4

<b>Managing Feeds .....</b>	<b>33</b>
Administering Feeds .....	33
Migrating Feeds Between Databases .....	33
Using SysAudit Information .....	35
Archiving Feeds .....	35
Pages Used to Archive Feeds .....	35
Archiving Scheduled Integration Broker Generic Message Feeds .....	35

## Chapter 5

<b>Publishing Integration Broker Generic Message Feeds .....</b>	<b>39</b>
Understanding Integration Broker Generic Message Feeds .....	39
Publishing Integration Broker Generic Message Feeds .....	40
Editing Integration Broker Generic Message Feeds .....	42
Deleting Integration Broker Generic Message Feeds .....	42

## Chapter 6

<b>Creating and Using Query Feeds .....</b>	<b>45</b>
Understanding Query Feeds .....	45
Publishing Query Feeds .....	46
Understanding Steps to Publish a Query Feed Definition .....	47
Publishing and Managing Query Feed Definitions .....	47
Defining Advanced Feed Options for Query Feeds .....	48
Using the Advance Feed Options Page for Query Feeds .....	49
Mapping Feed Entry Elements to Entry Templates .....	51
Using Mapping Builder to Edit Entry Templates .....	56
Using Query Feeds .....	61
Managing Query Feeds .....	62

## Chapter 7

<b>Creating and Using Worklist Feeds .....</b>	<b>63</b>
Understanding Worklist Feeds .....	63
Publishing Worklist Feeds .....	65
Understanding the Steps to Publish a Worklist Feed .....	65
Defining Worklist Feeds .....	65
Defining Worklist Advanced Feed Options .....	67
Using Worklist Feeds .....	70
Managing Worklist Feeds .....	71

## Chapter 8

<b>Developing New Feed Data Types .....</b>	<b>73</b>
Analyzing Requirements for New Feed Data Types .....	73
Creating the Feed Data Source Application Class .....	74
Extending the Base Class .....	74
Implementing the Methods .....	74
Setting Flags by Using Protected Methods .....	76
Defining the Feed Data Type .....	76
Page Used to Define Feed Data Types .....	76
Steps for Defining Feed Data Types .....	76
Defining Feed Data Types .....	77
Defining Advanced Options .....	79
Updating the Property Maintenance Component .....	79
Adding the Four Standard Publish as Feed Pages .....	80

Adding the Publish as Feed Link to a Page .....	80
Creating an Advanced Options Page .....	82
Adding Record PeopleCode .....	83
Updating the View Content Component or Pagelet .....	86
Adding the Related Feeds Hover Menu to Pages .....	86
Adding Component or Page PeopleCode .....	87
Adding the Related Feeds Hover Menu to Pagelet Wizard Pagelets .....	88
Developing Up-front Scheduled Feeds .....	89
Up-front Scheduled Feeds Prerequisites .....	89

## **Appendix A**

<b>Troubleshooting .....</b>	<b>91</b>
Common Problems .....	91

## **Appendix B**

<b>Configuring Integration Broker to Work With the Feed Publishing Framework .....</b>	<b>93</b>
Configuring the IB Gateway and Node Application Server .....	93
Configuring the Service Target Location .....	94
Configuring the Default User of the ANONYMOUS Node .....	94
Configuring the Content URI and Portal URI Node Properties .....	94

## **Appendix C**

<b>Configuring Oracle WebLogic Servers for Basic Authentication .....</b>	<b>97</b>
Modifying the config.xml File .....	97

<b>Index .....</b>	<b>99</b>
--------------------	-----------

# Oracle's PeopleSoft Feed Publishing Framework Preface

This preface discusses.

- PeopleTools products.
- About these PeopleBooks.

---

## PeopleTools Products

This PeopleBook refers to the following PeopleSoft Enterprise PeopleTools:

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

## Feed Publishing Framework

The Feed Publishing Framework consists of 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; and multiple language support.

## Integration Broker

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

## Query Manager

Query Manager 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.

## 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 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 Online PeopleSoft Library

A companion PeopleBook called PeopleBooks and the Online PeopleSoft 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 PeopleSoft online library including full-text searching and configuring a reverse proxy server.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Glossary of useful PeopleSoft terms that are used in PeopleBooks.

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

## Chapter 1

# Getting Started With PeopleSoft Feed Publishing Framework

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

---

## PeopleSoft Enterprise 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.

---

## 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 PeopleSoft 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.

---

## PeopleSoft Feed Publishing Framework Implementation

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

- Implement existing feeds.
- Develop and implement new feed data types.

## Prerequisites

The Feed Publishing Framework uses PeopleSoft Integration Broker Framework. The system must be configured as follows:

Step	Reference
Integration Broker must be configured and active.	<p>See <i>Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework</i>, "Configuring Integration Broker to Work With the Feed Publishing Framework."</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Integration Broker Administration</i>, "Using the Integration Broker Quick Configuration Page," Accessing the Integration Broker Quick Configuration Page.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Integration Broker Administration</i>, "Configuring PeopleSoft Integration Broker for Handling Services," Setting Service Configuration Properties.</p>
The default local node must have Authentication set to Password or Certificate, the Content URI text and Portal URI text of the default local node and any portal hosting nodes must be defined, and the default user of the ANONYMOUS mode must be a valid user.	<p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies</i>, "Configuring the Portal Environment," Setting Portal Nodes.</p>

<b>Step</b>	<b>Reference</b>
For Oracle WebLogic servers, you must configure the server for basic authentication.	See <i>Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework</i> , "Configuring Oracle WebLogic Servers for Basic Authentication."
Assign users the roles and permission necessary to use feeds.	See <i>Enterprise PeopleTools 8.50 PeopleBook: Security Administration</i> , "Setting Up Permission Lists," Granting Access to Components and Pages.  See <i>Enterprise PeopleTools 8.50 PeopleBook: Security Administration</i> , "Setting Up Roles," Assigning Permissions to Roles.  See <i>Enterprise PeopleTools 8.50 PeopleBook: Security Administration</i> , "Administering User Profiles," Setting Roles.
Activate Pub/Sub servers in application server.	See <i>Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration</i> , "Using PSADMIN Menus," Accessing the Application Server Options.
Configure Process Scheduler.	See <i>Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration</i> , "Using PSADMIN Menus," Creating a Process Scheduler Server Configuration.  See <i>Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration</i> , "Using PSADMIN Menus," Starting a Process Scheduler Server.

## Using PeopleTools-Delivered Feed Data Types

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

<b>Step</b>	<b>Reference</b>
Publish list of feeds for a data type.	See <i>Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework</i> , "Creating and Using Feeds," Publishing a List of Feeds Feed.
Create Generic IB Message feeds.	You can create generic IB message feeds from the Define IB Generic Message Feeds page.  See <i>Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework</i> , "Publishing Integration Broker Generic Message Feeds."
Create Worklist feeds.	You can create worklist feeds from the Worklist and Worklist Details pages.  See <i>Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework</i> , "Creating and Using Worklist Feeds."

<b>Step</b>	<b>Reference</b>
Create Query feeds.	You can create Query feeds from Query Manager pages. <i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Creating and Using Query Feeds."</i>

## Developing New Feed Types

Use these steps to develop new feed data types:

<b>Step</b>	<b>Reference</b>
Analyze requirements for new feed data type.	<i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types," Analyzing Requirements for New Feed Data Types.</i>
Implement feed data source application class.	<i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types," Creating the Feed Data Source Application Class.</i>
Define feed data type.	<i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types," Defining the Feed Data Type.</i>
Update the property maintenance page to include the Publish as Feed link.	<i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types," Adding the Publish as Feed Link to a Page.</i>
Add standard Publish as Feed pages to a component as hidden pages.	<i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types," Adding the Four Standard Publish as Feed Pages.</i>
Update the view content page to include the related feeds hover menu.	<i>See Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types," Adding the Related Feeds Hover Menu to Pages.</i>

## Chapter 2

# Understanding the Feed Publishing Framework

This chapter discusses:

- Feed Publishing framework.
- Feed data types.
- Real-time and scheduled feeds.
- 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 (IB) components.

Light blue boxes - Portal components or GUI.

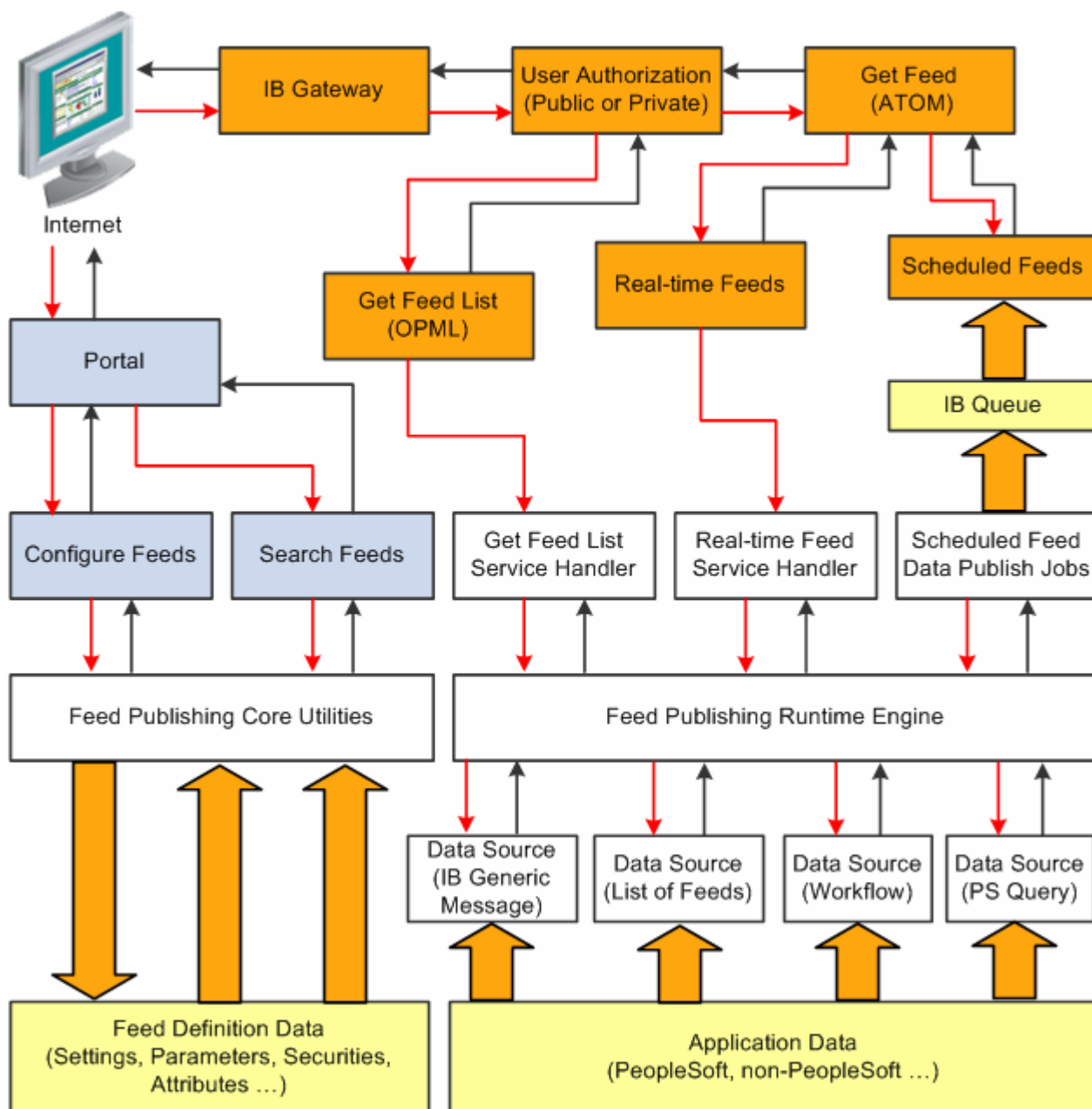
Yellow boxes - Data storage.

White boxes - Feed framework core.

Red lines - Incoming request.

Black lines - Response.

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.

- 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. The runtime engine also evaluates the runtime values of data source parameters, for which their Usage Types are *System Variable* and *User Specified*, at this time.

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 IB 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. IB 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 IB 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*.

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:

<b><i>Atom 1.0 Feed-Level Element</i></b>	<b><i>Feed Definition Property</i></b>
author	Author
category	Feed data type
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.

<i>Atom 1.0 Entry-Level Element</i>	<i>Item Property</i>
author	Author
category	Category <b>Note.</b> This item allows multiple entries.
content	FullContent, if available
contributor	Contributor <b>Note.</b> This item allows multiple entries.
id	GUID, if available or content URL
link	Alternate link: contentURL Other links: enclosure, if available. <b>Note.</b> 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.

### ***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.

**See Also**

*Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Developing New Feed Data Types"*

---

## Feed Data Types

This section discusses the four delivered feed data types:

- List of feeds (FEED).
- Integration Broker Generic message feeds (GENERICFEED).
- PS Query feeds (PSQUERY).
- Worklist feeds (WORKLIST).

### List of Feeds (FEED)

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

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Creating and Using Feeds,"* Publishing a List of Feeds Feed.

### Integration Broker Generic Message Feeds (GENERICFEED)

Integration Broker (IB) Generic message feeds enable administrators to expose IB messages used in asynchronous-one way service operations as feeds.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Publishing Integration Broker Generic Message Feeds."*

### PS Query Feeds (PSQUERY)

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

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Creating and Using Query Feeds."*

### Worklist Feeds (WORKLIST)

Worklist feeds enable workflow administrators to expose worklists as feeds.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Creating and Using Worklist Feeds."*

---

## Real-Time and Scheduled Feeds

This section discusses:

- Real-time feeds.
- Scheduled feeds.

### Real-Time Feeds

Real-time feeds are dynamic; produced when the user requests them. Real time feeds are created by using Integration Broker (IB) synchronous service operations. These service operations are similar to other IB 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 the 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 ATOM\_1\_0 response message.

### Scheduled Feeds

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

Scheduled feeds can be further classified into up-front feeds and generic feeds. When using up-front feeds, the messages published to the IB queues are feed messages. When using generic feeds, the messages published to IB 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 IB 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 IB 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.

---

---

## Feed Security

This section discusses security for:

- Creating feed data types.
- Publishing feeds.

- 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 *Enterprise PeopleTools 8.50 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).
- IB generic message feed pages. (PTFP\_GENERIC\_FEED).

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

To publish a list of feeds for each data type, you must be authorized to access pages in the PTFP\_DATATYPE component on the PTFP\_FEED\_PUBLISHING menu.

## 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 *and* whether the user has access to data in the feed when the user requests it. You can configure feed security to be:

- Public.

Public feeds use the default user that is associated with the ANONYMOUS node.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Integration Broker Administration*, "Adding and Configuring Nodes," Defining Node Parameters.

- 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 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).
- Publish as Feed (PTFP\_PUB\_AS\_LIST).
- Advanced Feed Options (PTFP\_PUB\_AS\_ADVOPT).
- 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 *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Feeds," Publishing Feeds.

---

## **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 file and save the list for later use.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Feeds," Using the My Feeds Page.



## Chapter 3

# 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 business process flow for feeds and discusses how to:

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

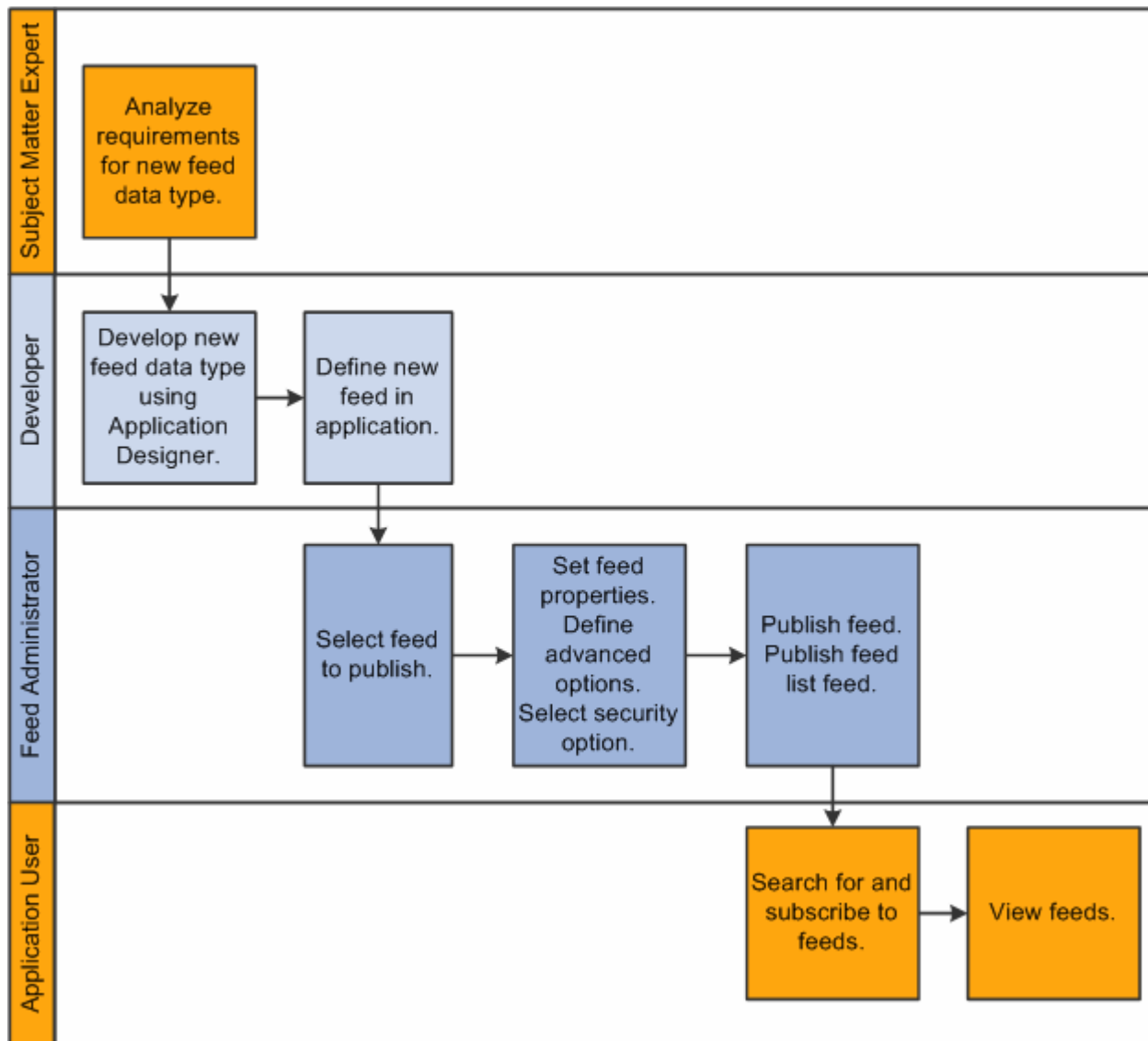
## Pages Used to Publish Feeds

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Publish Feed Definition	PTFP_PUB_AS_FEED	Click the Publish as Feed link from a data type content page.	Define feed security options, enter optional feed properties, and access advanced options.
Advanced Feed Options	PTFP_PUB_AS_ADVOPT	Click the Advanced Options link from the Publish Feed Definition page.	Enter advanced option values that are specific to the feed.
Publish as Feed	PTFP_PUB_AS_LIST	Click the Publish button from the Publish Feed Definition page.	Review, edit, or add feed definitions for this item.

Page Name	Definition Name	Navigation	Usage
Publish Feed Definition to Sites	PTFP_PUB_AS_SITES	Click the Publish Feed to Other Sites link from the Publish Feed Definition page.	Select additional sites to publish the feed.

## Feed Publishing Business Process Flow

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



Business process flow for developing, administering, and viewing feeds

## Publishing Feeds

To publish the feed, you access the Publish as Feed pages.

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

<b><i>Feed Data Type</i></b>	<b><i>Navigation</i></b>
WORKLIST	Worklist, Worklist or Worklist, Worklist Details
PSQUERY	Reporting Tools, Query, Query Manager
GENERICFEED	PeopleTools, Feeds, Define IB Generic Message Feed
FEED	PeopleTools, Feeds, Define Feed Data Types

## Defining Feed Properties

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

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

## Define Generic Feeds

### Publish Feed Definition

Set the values to create or update a feed definition.

Feed Properties

\*Feed Title: IB\_GENERIC

Description: Generic Operation

Advanced Options

Additional Feed Properties

Copyright:

Logo:

Icon:

Author Name: Robert Dupont

Author Email: rob.dupont@mydomain.com

Contributors

Customize | Find | First 1 of 1 Last

	Contributor Name	Contributor Email		
1	Irina Hergovich	irina.hergovich@mydomain.com	+	-

Feed Security Options

☐ Publish as Public
 ☒ Publish with Selected Security

☐ Publish with Realtime Security
 

Sync with Data

Selected Security

Customize | Find | First 1 of 1 Last

	*Type	Name	Description		
1	Permission List	QEIBUSR	QEIBUSR	+	-

Publish

Cancel

[Preview Feed](#)
[Publish Feed to Other Sites](#)

Publish Feed Definition page

20

Copyright © 1988, 2009, Oracle and/or its affiliates. All Rights Reserved.

## Feed Properties

<b>Feed Title</b>	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.
<b>Description</b>	Enter no more than a 255-character description of the feed.
<b>Advanced Options</b>	Click the Advanced Options link to access the Advanced Options page for the data type. Advanced options vary by feed data type.
<b>Publish</b>	Click the Publish button to publish the feed.

## Additional Feed Properties

Not all readers display all properties. This table 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 *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Understanding the Feed Publishing Framework," Feed Document Properties.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Developing New Feed Data Types," Defining the Feed Data Type.

<b>Copyright</b>	Enter a copyright date to be included in the XML.
<b>Logo</b>	Enter a URL to the logo to be included in the XML, for example, <a href="http://peoplebooks.peoplesoft.com:8700/PSOL/pt/img/logo.gif">http://peoplebooks.peoplesoft.com:8700/PSOL/pt/img/logo.gif</a> .
<b>Icon</b>	Enter a URL to an icon to be included in the XML, for example, <a href="http://peoplebooks.peoplesoft.com:8700/PSOL/pt/img/icon.gif">http://peoplebooks.peoplesoft.com:8700/PSOL/pt/img/icon.gif</a> .
<b>Author Name</b>	Enter an author to be included in the XML.
<b>Author Email</b>	Enter an author email address to be included in the XML.
<b>Contributor Name</b>	Enter a contributor to be included in the XML.
<b>Contributor Email</b>	Enter a contributor email address to be included in the XML.

## Feed Security Options

This table describes the feed security options:

---

**Note.** Query feeds apply security by using the Advanced Options page.

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

---

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Worklist Feeds," Understanding Worklist Feeds.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Query Feeds," Defining Advanced Feed Options for Query Feeds.

<b>Publish as Public</b>	Select to make the feed available for public access.
<b>Publish with Selected Security</b>	<p>Select to make the feed available based on the viewer's role-based permission lists.</p> <p>When you select this type of security, the Selected Security grid appears, enabling you to set role and permission list security.</p> <hr/> <p><b>Note.</b> User-based permission lists, such as the Primary Permission List, are not applied with this type of security.</p> <hr/>
<b>Publish with Realtime Security</b>	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.
<b>Sync with Data</b>	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.

### ***Selected Security***

<b>Type</b>	Select either <i>Role</i> or <i>Permission List</i> security.
<b>Name</b>	Enter the name of the permission list or role that has access to the feed.

## **Defining Advanced Options**

Advanced options differ by data type.

Access the Generic IB Message Feed Advanced Feed Options page.

## Define Generic Feeds

### Advanced Feed Options

Specify the advanced options of this feed.

Feed Title: DELETE\_ROLE

#### Feed Options

Max Number of Entries:  (Enter 0 for unlimited number of entries.)

Define Generic IB Message Feeds - Advanced Feed Options page

**Max Number of Entries** Enter the maximum number of entries that the feed should return to the user. Enter 0 for unlimited entries. The framework restricts the output to 300 entries.

**Note.** This value (300 entries) can be configured in the PTFP\_MAX\_ROW\_LIMIT field in the PS\_PTFP\_OPTIONS table. If the Max Number of Entries specified is 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 the PTFP\_MAX\_ROW\_LIMIT is 0 or does not exist, then the output is limited by Max Number of Entries.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Creating and Using Query Feeds,"* Defining Advanced Feed Options for Query Feeds

*Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework, "Creating and Using Worklist Feeds,"* Defining Worklist Advanced Feed Options

## Publishing Feeds to Additional Sites


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

Access the Publish Feed Definition to Sites page:

## Publish Feed Definition to Sites

Select the sites to publish the feed.

Feed Title: User Profile Feed

Target Sites			
Find   View All    First 1-6 of 6 Last			
	Select	Site Name	Description
1	<input type="checkbox"/>	CUSTOMER	Customer-facing registry content
2	<input checked="" type="checkbox"/>	EMPLOYEE	Employee-facing registry content
3	<input type="checkbox"/>	MOBILE	Mobile registry content for Employee, Customer, or Supplier content
4	<input type="checkbox"/>	PARTNER	Partner-facing registry content
5	<input type="checkbox"/>	PS_SITETEMPLATE	PORTAL
6	<input type="checkbox"/>	SUPPLIER	Supplier-facing registry content

☒ [Select All](#) ☐ [Clear All](#)

[Publish Feed](#) [Cancel](#)

Publish Feed Definition to Sites page

### Target Sites

<b>Select</b>	Select this check box to publish the feed definition to this site.
<b>Site Name</b>	Displays the valid sites to which you can publish the feed. This field is display only.
<b>Publish Feed</b>	Click to publish the feed to the additional sites.

---

## Publishing a List of Feeds Feed

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

To publish a list of feeds feeds:

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

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

- *FEED*

Select this option to produce a list of all lists of feeds to which the user has access.

- *GENERICFEED*

Select this option to produce a list of all generic IB message feeds to which the user has access.

- *PSQUERY*

Select this option to produce a list of all query feeds to which the user has access.

- *WORKLIST*

Select this option to produce a list of all worklist feeds to which the user has access.

3. Click the Publish as Feed link.

4. Click the feed definition to view the feed.

See [Chapter 3, "Creating and Using Feeds," Defining and Publishing Feeds, page 17.](#)

---

## Accessing Feeds

You can access feeds by using:

- My Feeds page.
- Related feeds hover menu.
- Get Feed List service operation.

## Using the My Feeds Page

Access the My Feeds page.

## My Feeds

Search and view published feeds.

▼ Additional Instructions

Use the criteria to narrow the search for a feed. The Keyword criteria filters the results for the Feed ID, Title, and/or Description that contains the entered value.

▼ Search Feed Definitions

\*Data Type:

All Data Types

▼

Search

Reset


Keyword:









Search In:

☒ Current Site

☐ All Sites

Feed Definitions

View All |  First 1-8 of 8 Last

	Feed Title
1	 <a href="#">Aggregate Feeds</a>
2	 <a href="#">CM_DIM_CTRL_TBL</a>
3	 <a href="#">CM_ATTRIBUTES</a>
4	 <a href="#">DELETE_ROLE</a>
5	 <a href="#">PS Query Feed</a>
6	 <a href="#">MESSAGE_SETS</a>
7	 <a href="#">QAS_TST_MSG_ITEMS</a>
8	 <a href="#">QAS_TST_MSG_ITEMS</a>

[Export Feed List](#)

My Feeds page

### Additional Instructions

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

### Search Feed Definitions

- Data Type

Select from the following feed data types:  
*List of Feeds* returns a list of all feed list feeds to which the user has access.  
*Query Feeds* returns a list of query feeds to which the user has access.  
*Generic Feeds* returns a list of Integration Broker generic feeds to which the user has access.  
*Worklist Feeds* returns a list of worklist feeds to which the user has access.  
*All Data Types* returns a list of all feeds to which the user has access.
- 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 sites to search for feeds:  
  
Current Site: Select this option to search in the viewer's current site only. This value is the default.  
  
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 enter.
- Reset

Click the Reset button to clear all fields, clear the feed list, and return to a blank search.
- Export Feed List

Click this link to export the current feed list in the Feed Definitions grid in a \*.opml file of OPML 2.0 format. OPML files can be imported in a feed reader, and the user will be able to subscribe to the list of feeds in the exported file.  
  
This link is visible only when a search returns one or more values.

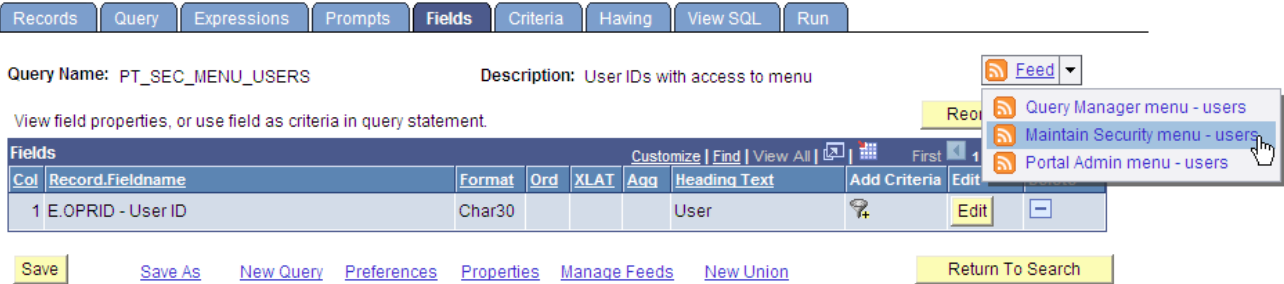
**Feed Definitions**

This grid lists the feeds that match the search criteria and that the viewer has permissions to view.

**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:



Query - Fields page showing the Related Feeds hover menu

**Note.** The hover menu is inactive when no feeds are available for that particular feed data type. Inactive hover menus can be hidden if you set the PTFP\_SHOW\_IA\_MENU field to 0 in the PS\_PTFP\_OPTIONS table.

## Using the Get Feed List 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.

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:

<b>Parameter</b>	<b>Description</b>	<b>Example</b>
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 <i>User</i> 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_NAME <sub>n</sub>	The data source setting name, where <i>n</i> is an integer beginning with 1.	To specify the first data source setting name, use this parameter: PTFP_DSS_NAME1=QRYNAME .
PTFP_DSS_VALUE <sub>n</sub>	The data source setting value, where <i>n</i> 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.** PeopleSoft PeopleTools provides one Get Feed List service. You may 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:

```

<?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"=>
/>
    <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&PORTAL_NAME=EMPLOYEE&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&PORTAL_NAME=EMPLOYEE&NODE_NAME=QE_LOCAL"/>
</body>
</opml>
```

---

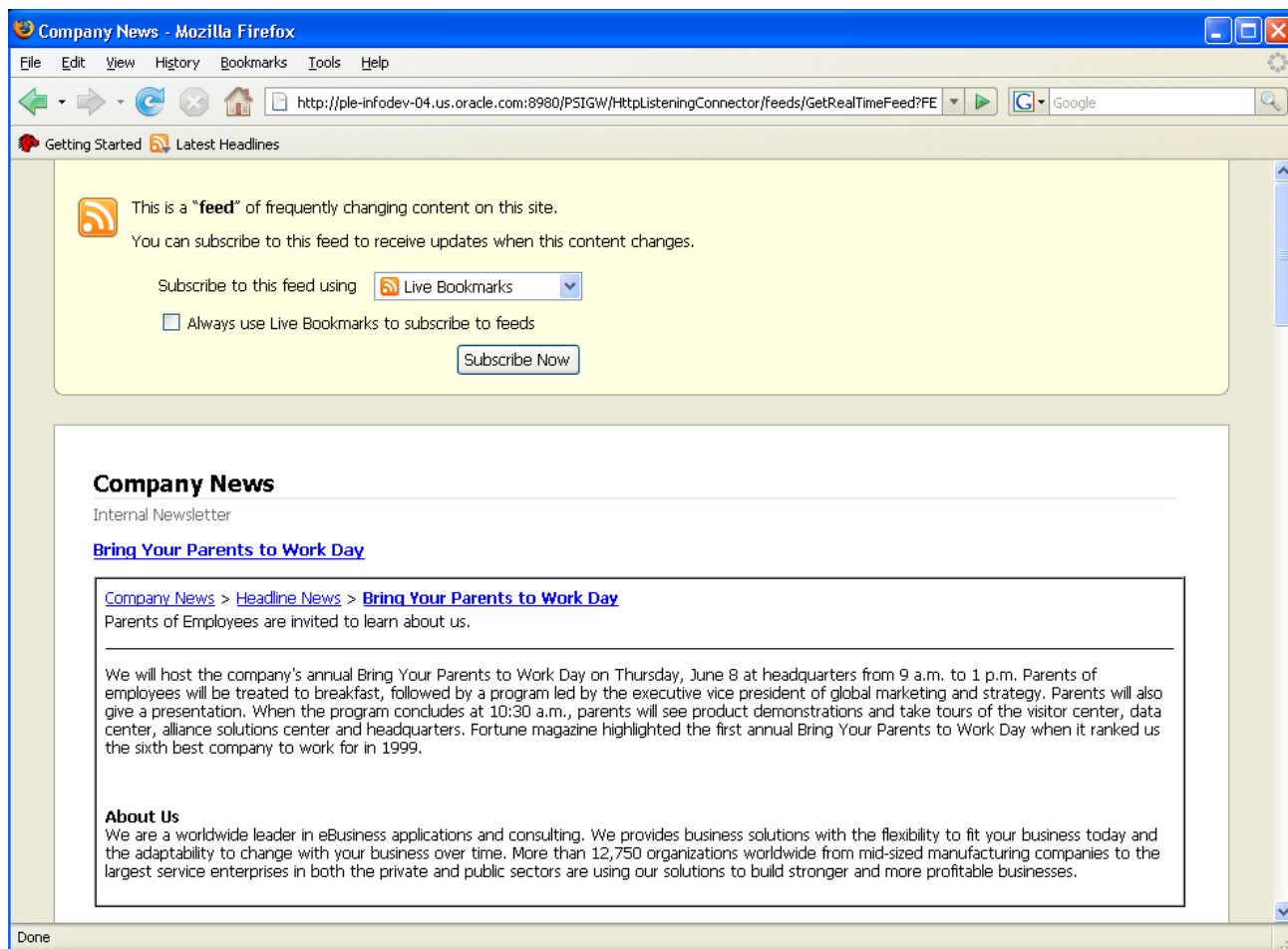
## Viewing Feeds

You can view feeds directly in the browser or by using a 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:



Company News feed in browser



## Chapter 4

# Managing Feeds

This chapter discusses how to:

- Administer feeds.
- Migrate feeds between databases.
- Use SysAudit information.
- Archive feeds.

---

## Administering Feeds

In a production 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 from the content maintenance page. The framework only provides the user interface and defines the behaviors of the pages. Application developers should add these pages to the content maintenance component and implement the code to handle that type of content feed creation using the Feed Publishing Core APIs.

---

## Migrating Feeds Between Databases

Use Data Mover to migrate feeds between databases.

See *Enterprise PeopleTools 8.50 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.
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**

*Enterprise PeopleTools 8.50 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**

*Enterprise PeopleTools 8.50 PeopleBook: Data Management*, "Ensuring Data Integrity," Running SYSAUDIT

## Archiving Feeds

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

### Pages Used to Archive Feeds

Page Name	Definition Name	Navigation	Usage
Run Feed Archive	IB_FEEDARCHIVE	PeopleTools, Feeds, Archive Feed Data	Archive Generic IB messages used by scheduled feeds.
Process Scheduler Request	PRCSRQSTDLG	PeopleTools, Feeds, Archive Feed Data	Archive Generic IB messages used by scheduled feeds.

### Archiving Scheduled Integration Broker Generic Message Feeds

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

#### Run Feed Archive

Run Control ID: ARCH\_FEED\_1

Report Manager

Process Monitor

Run

Run Feed Archive page

Click the Run button to access the PTIBFEEDARCH process:

### Process Scheduler Request

User ID: QEDMO Run Control ID: ARCH\_FEED\_1

---

Server Name:  Run Date: 04/20/2009

Recurrence:  Run Time: 9:57:12AM Reset to Current Date/Time

Time Zone:

Select	Description	Process Name	Process Type	*Type	*Format	Output Destination
<input checked="" type="checkbox"/>	PTIBFEEDARCH	PTIBFEEDARCH	Application Engine	File	PDF	c:\TEMP

Process Scheduler Request page showing the PTIBFEEDARCH process

Use the Archive Feeds page to archive IB messages that are used by scheduled feeds. Archiving takes no parameters from the user; parameters for archiving come from the feed definitions.

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

You can view archived IB 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 IB 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 *Enterprise PeopleTools 8.50 PeopleBook: Integration Broker Service Operations Monitor*, "Monitoring Asynchronous Service Operations," Monitoring Asynchronous Service Operation Transactions.

### Feed Archiving Parameters

The following parameters specified in the feed definition are used for archiving the IB messages:

- IB Service Operation name

For generic 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.

- IB sub queue name, when applicable

For Generic feeds, no sub queue 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 IB queue appears to the user.
- If the parameter value is *PTFP\_FEED:UTILITY:Utility SF\_MAXROWOPTION\_ALLMSGs*, then all the messages in the IB 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 IB 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 IB queue for fewer than 100 minutes appear to the user.

### ***Feed Archiving Options and Logic***

This table describes the feeds archiving options and logic:

<b><i>Option Value</i></b>	<b><i>Description</i></b>	<b><i>Archiving Logic</i></b>
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 sub queue per language.	When DSPARAMETER_SF_MAXMINUTES is SF_MAXMINUTES_ALLMSGs and DSPARAMETER_MAXROW is SF_MAXROWOPTION_LATESTMSG.
2	Archive all messages except latest <i>n</i> messages per sub queue 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 <i>not</i> 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 feed definition. These cannot be specified when the Archive Feed Application Engine program runs.

---

## Chapter 5

# Publishing 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.
- Edit Integration Broker generic message feeds.
- Delete Integration Broker generic message feeds.

---

## Understanding Integration Broker Generic Message Feeds

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

The Define IB Generic Message 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 shows Local-to-Atom routing for an operation:

General

Handlers

Routings

Service Operation: USER\_PROFILE  
Operation Type: Asynchronous - One Way  
\*Operation Description: User Profile  
Operation Comments:   
☐ User/Password Required  
\*Security Verification: None  
Object Owner ID: PeopleTools  
Operation Alias:   
[Service Operation Security](#)

Default Service Operation Version

\*Version: VERSION\_84  
Version Description: User Profile  
Version Comments:   
☐ Non-Repudiation  
☐ Runtime Schema Validation  
[Intropection](#)

☒ Default ☒ Active  

Routing Status  
Any-to-Local: Does not exist  
Local-to-Local: Does not exist  
Local-to-Atom: Does not exist.

Routing Actions Upon Save  
☐ Generate Any-to-Local  
☐ Generate Local-to-Local

Message Information

Type: Request  
Message.Version: USER\_PROFILE.VERSION\_84 [View Message](#)  
\*Queue Name: USER\_PROFILE [View Queue](#) [Add New Queue](#)

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.

The default and recommended security option for generic IB 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 security page to access IB generic message feeds. Users in the PeopleSoft Administrator and Portal Administrator roles also have access to IB generic message feeds.

## Publishing Integration Broker Generic Message Feeds

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

## Define Generic Feeds

### Define Generic Feeds

Define feeds for Generic Integration Broker messages.



Service Operation: USER\_PROFILE  
Operation Description: User Profile

[Publish as Feed](#)

Define Generic Feeds page

To publish Integration Broker generic message feeds:

1. Select PeopleTools, Feeds, Define IB Generic Message Feeds.
2. Select the service operation to publish as a feed.
3. Click the Publish as Feed link.
4. Change default feed properties if necessary.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Feeds," Defining Feed Properties.

5. Select the appropriate security option.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Understanding the Feed Publishing Framework," Feed Security.

6. Click the Advanced Options link.

The Advanced Feed Options page appears.

## Define Generic Feeds

### Advanced Feed Options

Specify the advanced options of this feed.

Feed Title: User Profile Feed

#### Feed Options

Max Number of Entries:  (Enter 0 for unlimited number of entries.)

Define Generic Feeds - Advanced Feed Options page

7. Enter a value for the maximum number of entries and click the OK button.

8. Click the Publish button.

---

## Editing Integration Broker Generic Message Feeds

Access the Define Generic Feeds - Publish as Feed page (click the Publish as Feed link from the Define Generic Feeds page).

### Define Generic Feeds

#### Publish as Feed

Review, edit or add feed definitions for this item. Only feed definitions published in the current site are marked as published and can be edited.

Feed Definitions			
		First	Last
	1 of 1		
	Feed Title	Published	
1	 <a href="#">USER PROFILE</a>	<input checked="" type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>

Define Generic Feeds - Publish as Feed page

To edit IB generic message feeds:

1. Click the Edit button.

The Publish Feed Definition page appears.

2. Click the Advanced Options link to edit advanced options.

The Advanced Feed Options page appears.

3. Edit advanced options if necessary.
4. Click the OK button.
5. Enter values in the Additional Feed Properties group box, as necessary, to override the default properties.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Feeds," Defining Feed Properties.

6. Select the appropriate security option.

See *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Understanding the Feed Publishing Framework," Feed Security.

7. Click the Publish button.

---

## Deleting Integration Broker Generic Message Feeds

Access the Define Generic Feeds - Publish as Feed page (click the Publish as Feed link from the Define Generic Feeds page).

Define Generic Feeds

Publish as Feed

Review, edit or add feed definitions for this item. Only feed definitions published in the current site are marked as published and can be edited.


Feed Definitions

123

First

1 of 1

Last

	Feed Title	Published	
1	 <a href="#">USER PROFILE</a>	<input checked="" type="checkbox"/>	<div>EditDelete</div>

Define Generic Feeds - Publish as Feed page

To delete an IB generic message feed:

1. Select the service operation to use for the feed.
2. Click the Publish as Feed link.
3. To delete the feed, click the Delete button for the appropriate feed.
4. Click OK to confirm the delete.



## Chapter 6

# Creating and Using Query Feeds

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

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

---

## Understanding Query Feeds

This section discusses query feeds.

### ***Query Feed Data Types***

PeopleSoft delivers the Query feed data type, which uses the application class created for query feeds.

Access the Define Feed Data Types page (select PeopleTools, Feeds, Define Data Types, PS Query (PTFP\_DATATYPE)).

Define Feed Data Types

Define Feed Data Types

Define the data types used by the Feed Publishing Wizard

Feed

Data Type:

PSQUERY

\*Description:

PS Query Feeds

☒ Active

Long Description:

A feed of this data type contains data of the specified PS Query.

Default Feed Properties

Supporting Application Class

Package Name:

PTFP\_FEED



Path:



DataSource



Application Class ID:

PSQueryDataSource

Feed Service Operations

Customize | Find |  | 

First  1-2 of 2  Last

	*Service Operation	Description	Type	Default		
1	<div>PT_DYNAMIC_QU</div>	RealtimeQueryFeed	Real Time	<input checked="" type="checkbox"/>	<div><div>+</div></div>	<div><div>-</div></div>
2	<div>PT_PREPUBLISH</div>	ScheduledQueryFeed	Scheduled	<input type="checkbox"/>	<div><div>+</div></div>	<div><div>-</div></div>

[Publish as Feed](#)

Define Feed Data Types page showing PSQUERY data type

Query Feed Security

There are two levels of security in core Feed Publishing Framework, feed security and data security. Feed security controls who can see the feed when searching it. Data security controls whether the underlying data can be seen in the feed entry when user request it at runtime.

Query feed security, which users can specify in advance options page in feed designer, is for defining feed security.

Publishing Query Feeds

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

## Understanding Steps to Publish a Query Feed Definition

Any user with access to Query Manager will be able to publish query feeds.

Complete these steps to publish a query feed definition:

1. Select Reporting Tools, Query, Query Manager.
2. Select the query that you want to publish and click the Edit link.
3. Click the Publish as Feed link.
4. Enter a feed title and description.

See Chapter 6, "Creating and Using Query Feeds," Publishing and Managing Query Feed Definitions, page 47.

5. Enter overrides for Additional Feed Properties and Contributors.
6. Click the Advanced Options link.

See Chapter 6, "Creating and Using Query Feeds," Defining Advanced Feed Options for Query Feeds, page 48.

7. Enter the query parameters and values for the feed.
8. Click OK.
9. Click Publish.

## Publishing and Managing Query Feed Definitions

Access Query Manager (Reporting Tools, Query, Query Manager).

1. Select the query to publish.
2. Click the Publish as Feed link.

The Publish Feed Definition page appears.

3. Click the Advanced Options link to map query data.

The Advanced Feed Options page appears.

See Chapter 6, "Creating and Using Query Feeds," Defining Advanced Feed Options for Query Feeds, page 48.

4. Set advanced options and change the feed entry content mapping values, if necessary.

See Chapter 6, "Creating and Using Query Feeds," Mapping Feed Entry Elements to Entry Templates, page 51.

5. Click the OK button.
6. Enter values in the Additional Feed Properties group box to override default properties.

7. Select the appropriate security option.
8. Click the Publish button.

### ***Managing Query Feeds***

Access Query Manager (Reporting Tools, Query, Query Manager).

To edit query feeds:

1. Select Reporting Tools, Query, Query Manager.
2. Select the query to publish.
3. Click the Manage Feeds link.

The Publish as Feed page appears.

4. Click the Edit button.

The Publish Feed Definition page appears.

5. Click the Advanced Options link to edit advanced options.

The Advanced Feed Options page appears.

See [Chapter 6, "Creating and Using Query Feeds," Defining Advanced Feed Options for Query Feeds, page 48.](#)

6. Edit advanced options and change the feed entry content mapping values if necessary.

See [Chapter 6, "Creating and Using Query Feeds," Mapping Feed Entry Elements to Entry Templates, page 51.](#)

7. Click the OK button.
8. Enter values in the Additional Feed Properties group box to override default properties.
9. Select the appropriate security option.
10. Click the Publish button.

---

## **Defining Advanced Feed Options for Query Feeds**

This section discusses how to:

- Use the Advanced Feed Options page for query feeds.
- Map feed entry elements to entry templates.
- Use Mapping Builder to edit entry templates.

## Using the Advance Feed Options Page for Query Feeds

Access the Advanced Feed Options page (select Reporting Tools, Query Manager and click the Publish as Feeds link). On the Publish Feed Definition page, click the Advanced Options link.

### PSQuery Data Type

## Advanced Feed Options

Specify the advanced options of this feed.

Feed Title: PTPT1200 (PeopleTools) Users

#### ▼ Query Prompts

Parameter ID	Description	Value
CLASSID	Permission List	PTPT1200

#### ▼ Advanced Query Feed Options

Max Number of Entries:  (Enter 0 for unlimited number of entries.) [Preview Feed](#)

##### Entry Occurrence

- ☐ All Rows in One Feed Entry
- ☒ One Row Per Feed Entry

##### Feed Publishing Type

- ☐ Scheduled
- ☒ Real Time

##### Publish Language

- ☐ Current Language Only
- ☒ All Available Languages

\*Query Feed Security  ▼

#### ► Query Fields

Feed Entry Content Mapping			Find	First	1-5 of 5	Last
	*Feed Entry Element	Entry Template				
?	Entry Content Url ▼	http://buffy.us.oracle.com:8920/psp/QEDMO/EMP				
?	Entry Full Content ▼	%ROW.HTML%				
?	Entry ID ▼	%GUID%				
?	Entry Title ▼	%ROW.TEXT%				
?	Entry Updated ▼	%DateTime%				

PS Query 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.

## Advanced Query Feed Options

**Max Number of Entries** Enter the maximum number of feeds entries to display in the feed reader. Enter 0 to return an unlimited number of rows.

**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 IB 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 *Enterprise PeopleTools 8.50 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.

**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 *Enterprise PeopleTools 8.50 PeopleBook: Feed Publishing Framework*, "Creating and Using Feeds," Feed Security Options.

**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 passes to the feed reader. 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.

---

**Note.** Click the Zoom icon to work with the Feed Entry Content Mapping grid in its own modal window or to work with each template entry in its own modal window.

---

See [Chapter 6, "Creating and Using Query Feeds," Using Mapping Builder to Edit Entry Templates, page 56.](#)

**Mapping Feed Entry Elements to Entry Templates**

The Feed Entry Content Mapping grid has these fields:

**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	<p>A template is a string that encloses bind variables inside % symbols. When a 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.</p> <hr/> <p><b>Note.</b> Click the Zoom icon to work with each entry in its own modal window.</p> <hr/> <p>Entry templates support these bind variable types:</p> <ul style="list-style-type: none"><li>• <i>System variable</i>, for example, %Copyright%.</li><li>• <i>Template variable</i>, for example, %ROW.HTML%.</li><li>• <i>Query columns</i>, for example, %ALIASNAME.QUERYCOLUMNNAME%.</li><li>• <i>Message Catalog text</i>, for example, %MSGT:10.10%.</li><li>• <i>Message Catalog explanation</i>, for example, %MSGE:10.10%.</li></ul>
----------------	--

**Feed Entry Content Mapping**

The Feed Entry Content Mapping grid has these fields:

Feed Entry Element	<p>Each feed entry element is a property of an entry that can be interpreted and displayed by a feed reader.</p> <hr/> <p><b>Note.</b> Not all feed readers display all elements. Consult the documentation on your feed reader.</p> <hr/>
Entry Template	<p>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.</p> <hr/> <p><b>Note.</b> Click the Zoom icon to work with each entry in its own modal window.</p> <hr/> <p>Entry templates support these bind variable types:</p> <ul style="list-style-type: none"><li>• <i>System variable</i>, for example, %Copyright%.</li><li>• <i>Template variable</i>, for example, %ROW.HTML%.</li><li>• <i>Query columns</i>, for example, %ALIASNAME.QUERYCOLUMNNAME%.</li><li>• <i>Message Catalog text</i>, for example, %MSGT:10.10%.</li><li>• <i>Message Catalog explanation</i>, for example, %MSGE:10.10%.</li></ul>

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

<b>Feed Entry Element</b>	<b>Description</b>	<b>Action of Feed Reader</b>	<b>Entry Template Values</b>	<b>Runtime Processing</b>
<i>Entry Title</i>	<p>This element is a brief, single-line explanation of the feed entry.</p> <p>Only one entry title can exist per entry.</p>	The reader displays the title as indicated by the template	<p>Any template containing any bind variable.</p> <p>Can have HTML tags but needs to be only one line</p>	The system escapes the HTML as well as XML characters.
<i>Entry Content URL</i>	<p>This element is the URL to the PeopleSoft application page or other content.</p> <p>Only one entry content URL can exist per entry.</p>	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.
<i>Entry Categories</i>	<p>This element defines the category of an entry.</p> <p>Multiple entries can belong to the same category and one entry can belong to multiple categories.</p>	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.
<i>Entry ID</i>	<p>This element is a permanent, universal identifier.</p> <p>Only one ID can exist per entry.</p>	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.
<i>Entry Updated</i>	<p>This element is the date and time that the entry was most recently modified.</p> <p>Only one update date can exist per entry.</p>	The reader uses this entry to determine, in conjunction with the ID, whether to update the data in the entry.	<p>Static value in PeopleTools date and time format.</p> <p>Dynamic value present in any date time query column.</p> <p>System variable %DateTime.</p>	Values that are PeopleTools date time formatted will be converted to atom specific date time format.

<b>Feed Entry Element</b>	<b>Description</b>	<b>Action of Feed Reader</b>	<b>Entry Template Values</b>	<b>Runtime Processing</b>
<i>Entry Published</i>	<p>This element is the date and time when the entry originally created.</p> <p>Only one publish date can exist per entry.</p>	<p>The reader uses this entry to determine the maximum age of the data.</p>	<p>Static value in PeopleTools date and time format.</p> <p>Dynamic value present in any date time query column.</p> <p>System variable %DateTime.</p>	<p>Values that are PeopleTools date time formatted will be converted to atom specific date time format.</p>
<i>Entry Author</i>	<p>This element pertains to information about the creator, owner, and author of this entry.</p> <p>Only one author can exist per entry.</p>	<p>The reader organizes and filters entries based on the value of this field.</p> <p>The reader can use this field to send email to author.</p>	<p>Static values in this format:</p> <p>Name [ Email⇒ Address ]</p> <p>Author name and email ID can be mapped to query columns or system variables.</p>	<p>The system extracts the email address extracts from [] and the name from the value before the brackets.</p> <p><b>Note.</b> The brackets may be left empty, but must be present for the system to correctly process this entry.</p> <p>The system escapes the HTML as well as XML characters.</p>
<i>Entry Contributors</i>	<p>This element pertains to information about various parties involved in this entry.</p> <p>Multiple contributors can exist per entry.</p>	<p>The reader organizes and filters entries based on the value of this field.</p> <p>The reader can use this field to send email to author.</p>	<p>Static values in this format:</p> <p>Name [ Email⇒ Address ]</p> <p>Contributor name and email ID can be mapped to query columns or system variables.</p>	<p>The system extracts the email address extracts from [] and the name from the value before the brackets.</p> <p><b>Note.</b> The brackets may be left empty, but must be present for the system to correctly process this entry.</p> <p>The system escapes the HTML as well as XML characters.</p>
<i>Entry Description</i>	<p>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</p>	<p>The reader presents more information than the title, but less than the full content while providing a link to the full content.</p>	<p>Any template containing any bind variable.</p> <p>HTML tags.</p>	<p>The system escapes the HTML as well as XML characters.</p>

<b>Feed Entry Element</b>	<b>Description</b>	<b>Action of Feed Reader</b>	<b>Entry Template Values</b>	<b>Runtime Processing</b>
<i>Entry Enclosures</i>	<p>This element contains other content types accessible using URL, for example images and mp3 files.</p> <p>Multiple enclosures signifying multiple attachments can exist per entry.</p>	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.	<p>URL to the resource, its content, and its length in bytes.</p> <p>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.</p>	The system escapes the HTML as well as XML characters.
<i>Entry Full Content</i>	<p>This element is the body of the entry.</p> <p>Only one full content can exist per entry.</p>	The reader can display the content of the entry by default or upon user request.	<p>Any template containing any bind variable.</p> <p>HTML tags.</p>	The system escapes the HTML as well as XML characters.
<i>Expires</i>	<p>This element is the date time when this entry is invalid and should be considered outdated.</p> <p>Only one expiration can exist per entry.</p>	The reader determines when to discard this entry.	<p>Static value in PeopleTools date and time format.</p> <p>Dynamic value present in any date time query column.</p>	<p>Values that are PeopleTools date time formatted will be converted to atom specific date time format.</p> <p>The system escapes the HTML as well as XML characters.</p>
<i>Max Age</i>	<p>This element is the length of time in milliseconds until when the entry will be valid after the Published or Updated date time.</p> <p>Only one maximum age can exist per entry.</p>	The reader determines when to discard this entry.	<p>Static number or mapped to a query column which results in number.</p>	The system escapes the HTML as well as XML characters.

<b>Feed Entry Element</b>	<b>Description</b>	<b>Action of Feed Reader</b>	<b>Entry Template Values</b>	<b>Runtime Processing</b>
<i>Entry Copyright</i>	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

Entry templates can be edited using Mapping Builder, which enables you to create custom output to the feed reader by building 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 users is not supported. However, you can use the Export to Excel icon to export the grid values and share the values among users and with other query feeds.

---

To access Mapping Builder, click the Edit icon associated with the entry element that you want to change.



Click the Edit icon to access Mapping Builder.



Click the Zoom icon to work with an entry in its own modal window.

### 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:

Mapping Builder

PSQuery Data Type

Advanced Feed Option - Feed Element Mapping Builder

Feed Element: Entry Author

Mapping Builder

\*Type: Query Fields

Variable: A.QE\_EMPLOYEE\_NAME

\*Type: Static Text

Text: [

\*Type: System Variables

Value: %EmailAddress

\*Type: Static Text

Text: ]

Preview

Entry Template:

%A.QE\_EMPLOYEE\_NAME%[%EmailAddress%]

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

The Mapping Builder fields are described as follows.

<b>Type</b>	<p>Select from these mapping builder types to assemble the entry template:</p> <p><i>Message Catalog Explanation</i></p> <p><i>Message Catalog Text</i></p> <p><i>Query Fields</i></p> <p><i>Static Text</i></p> <p><i>System Variables</i></p> <p><i>Templates</i></p>
<b>Message</b>	<p>This field is visible when the Type field value is <i>Message Catalog Explanation</i> and <i>Message Catalog Text</i>.</p> <p>Enter the message number to appear in the entry.</p> <hr/> <p><b>Note.</b> Use the Message Catalog rather than static text to generate a template that is language independent.</p> <hr/>
<b>Msg Set</b> (message set)	<p>Visible when the Type field value is <i>Message Catalog Explanation</i> and <i>Message Catalog Text</i>.</p> <p>Enter the message set number to appear in the entry.</p> <hr/> <p><b>Note.</b> Use the Message Catalog rather than static text to generate a template that is language independent.</p> <hr/>
<b>Text</b>	<p>This field is visible when the Type field value is <i>Static Text</i>.</p> <p>Enter the text that you want to appear in the entry.</p> <hr/> <p><b>Note.</b> To have a percent sign appear in the text of the template, it needs to be escaped using <code>%%</code>. For example, if the template reads <code>%%</code>, then at runtime the builder replaces it with <code>%</code>.</p> <hr/>
<b>Value</b>	<p>This field is visible when the Type field value is <i>System Variables</i>.</p> <p>Select from any of the available system variables.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference</i>, "System Variables."</p>
<b>Variable</b>	<p>This field is visible when the Type field value is <i>Templates</i> or <i>Query Fields</i>.</p> <p>When the Type field value is <i>Templates</i>, select from these two Template options:</p> <ul style="list-style-type: none"> <li>• <i>Row in HTML Format</i></li> <li>• <i>Row in TEXT Format</i></li> </ul> <p>When the Type field value is <i>Query Fields</i>, select from any field comprising the query definition.</p>

### ***Creating Entry Templates in the Rich Text Editor***

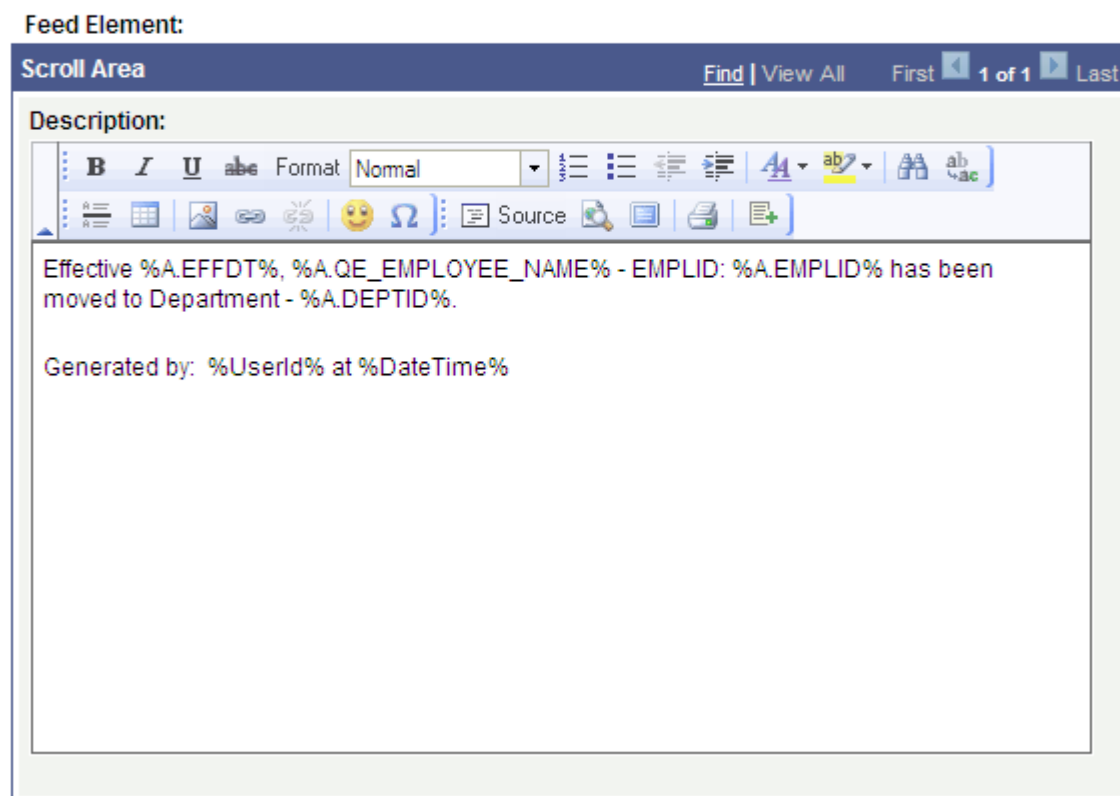
These feed entry elements present the rich text editor, where you can enter HTML, rich text, and bind variables to create a variety of output types to the feed reader:

- Entry Description
- Entry Full Content
- Entry Title

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

#### **Mapping Builder**

### **Advanced Feed Option - Feed Element Mapping Builder**



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

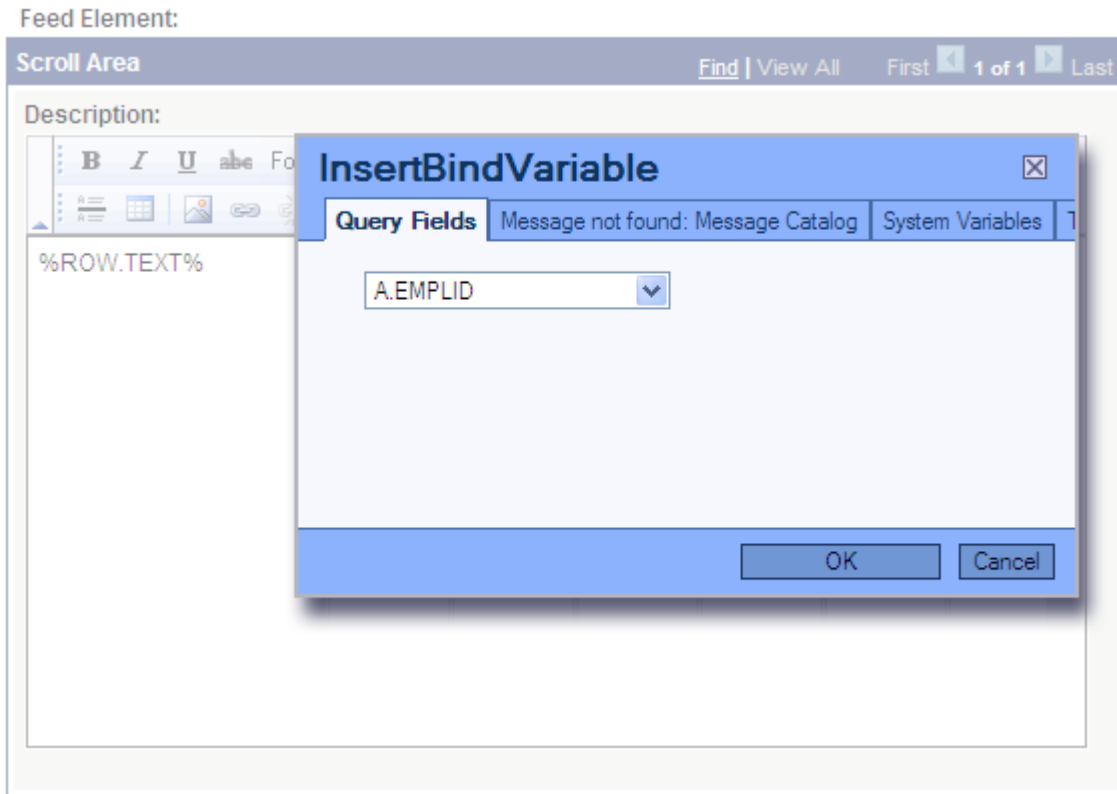
### ***Accessing Bind Variables in the Rich Text Editor***

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

- Query Fields

- 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

Query feeds are secured by user IDs. The query user must have access to the service operation PTFP\_FEED. This access is included in permission list *PTPT1000*, which is in the role *PeopleSoft User*. When a user is signed on to PeopleSoft Pure Internet Architecture, the user can view the feeds by selecting either of the following navigation paths:

- Select My Feeds.  
Enter search criteria and click Search.  
Select the desired query feed.
- Access Query Manager and use the Related Feeds hover menu.




# Managing Query Feeds

Access the Publish as Feed (PTFP\_PUB\_AS\_FEED) page (click the Publish as Feed link from Query Manager pages).

PSQuery Data Type

Publish as Feed

Review, edit or add feed definitions for this item. Only feed definitions published in the current site are marked as published and can be edited.

Feed Definitions		First	1-3 of 3	Last
	Feed Title	Published		
1	 <a href="#">Query Manager menu - users</a>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Delete</a>
2	 <a href="#">Maintain Security menu - users</a>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Delete</a>
3	 <a href="#">Portal Admin menu - users</a>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Delete</a>

Query Publish as Feed page

All of the published query feed definitions for the selected query appear. From this page, you can edit, delete, or add query feeds.

- Edit**

Use this button to open the Publish Feed Definition page, where you can edit the current published definition.
- Delete**

Use this button to delete the query feed.
- Add Feed**

Use this button to define and publish a new query feed.

## Chapter 7

# Creating and Using Worklist Feeds

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

- Publish worklist feeds
- Use worklist feeds
- Manage worklist feeds

---

## Understanding Worklist Feeds

Worklist feeds provide the ability to publish a user's worklist as a feed in real time. By using worklist feeds, the worklist items can be available from systems other than the system that originated the entries without copying the data. Worklist feeds use the Feed Publishing Framework for delivering feeds. PeopleTools delivers the WORKLIST feed data type to publish worklist feeds.

Worklist feeds are real-time feeds. The feed is generated in the security context of the signed on user and it reflects the user's current worklist items. It does not show worked items.

### ***Worklist Feed Additional Configuration***

In addition to all configuration steps for other feeds, worklist feeds require an additional step.

The node that contains the Worklist and Worklist Details pages should have Content URI Text and Portal URI Text defined. This text will be used for generating the worklist item URLs. To find the node name, navigate to the Worklist and Worklist Details pages and look at the URL. It will be as follows:

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

In the previous URL, PT\_LOCAL is the node, therefore, you should define the Content URI Text and the Portal URI Text for the PT\_LOCAL node so that worklist feeds point to the correct worklist items.

### ***WORKLIST Feed Data Type***


PeopleSoft Feed Publishing Framework delivers the WORKLIST feed data type for worklist feeds.

To access the Define Feed Data Types page, select PeopleTools, Feeds, Define Feed Data Types (PTFP\_DATATYPE).

Define Feed Data Types

Define Feed Data Types

Define the data types used by the Feed Publishing Wizard

 Feed ▾

Data Type:

WORKLIST

\*Description:

Worklist Feed

☒ Active

Long Description:

Feeds of this data type contain worklist items of the current user.

▼ Default Feed Properties

Copyright:



Logo:



Icon:

Author Name:

Author Email:

Contributors


Customize | Find |   First 1 of 1 Last

	Contributor Name	Contributor Email		
1				

Supporting Application Class


Package Name:

PTFP\_FEED




Path:

DataSource








Application Class ID:

WorklistDataSource



Feed Service Operations

Customize | Find |   First 1 of 1 Last

	*Service Operation	Description	Type	Default		
1	<div>PTFP_GETFEED</div>	Real-time feed operation	Real Time	<input checked="" type="checkbox"/>		

Define Feed Data Types page

The only values that should be changed for the WORKLIST data type are the Default Feed Properties.

Not all readers display all properties. The default feed properties that some feed readers process and display are described below.

**Note.** These default properties are for all worklist feeds. 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://peoplebooks.peoplesoft.com:8700/PSOL/pt/img/logo.gif*.

<b>Icon</b>	Enter a URL to an icon to be included in the XML, for example, <i>http://peoplebooks.peoplesoft.com:8700/PSOL/pt/img/icon.gif</i> .
<b>Author Name</b>	Enter an author to be included in the XML.
<b>Author Email</b>	Enter an author email address to be included in the XML.
<b>Contributor Name</b>	Enter a contributor to be included in the XML.
<b>Contributor Email</b>	Enter a contributor email address to be included in the XML.

---

## Publishing Worklist Feeds

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

- Define worklist feeds.
- Define worklist advanced feed options.

## Understanding the Steps to Publish a Worklist Feed

Any user with the role *PeopleSoft Administrator* will be able to publish worklist feeds from either the Worklist or Worklist Details pages.

Access the Worklist page (select Worklist, Worklist).

To publish a Worklist Feed:

1. Click the Publish as Feed link.
2. Enter a Feed Title and description.
3. Enter any overrides necessary for Additional Feed Properties and Contributors.
4. Click the Advanced Options link.
5. Enter the worklist parameters and values for the feed.
6. Click OK.
7. Click Publish.

## Defining Worklist Feeds

Access the Publish Feed Definition (PTFP\_PUB\_AS\_FEED) page from either the Worklist or Worklist Details pages by clicking the Publish as Feed link.

Worklist  
**Publish Feed Definition**

Set the values to create or update a feed definition.

Feed Properties

\*Feed Title:

Worklist Items

Description:

List of Worklist Items

[Advanced Options](#)

▼ Additional Feed Properties

Copyright:

Logo:

Icon:

Author Name:

Author Email:

Contributors

Customize | Find | First 1 of 1 Last

	Contributor Name	Contributor Email		
1				

Publish Feed Definition page

In Additional Feed Properties, enter feed properties that will override the properties set on the Data Feed Type page. You can also add contributors.

**Feed Properties**

<b>Feed Title</b>	<p>This field is prepopulated with Worklist Items. Change the title to a name that will help you identify the specific worklist.</p> <hr/> <p><b>Note.</b> Multiple worklists can be entered for one title; however, since the worklist parameters entered on the Advanced Options page use AND logic, you may not get the desired results.</p> <hr/>
<b>Description</b>	<p>Enter a description of the feed. No more than 255 characters are allowed.</p>
<b>Advanced Options</b>	<p>Click the Advanced Options link to access the Worklist Advanced Feed Options page.</p>
<b>Publish</b>	<p>Click the Publish button to publish the feed.</p>

- Preview Feed**

Use this link to preview the feed after defining the Advanced Options.
- Publish Feed to Other Sites**

Select this link to publish the worklist to other PeopleSoft sites.

Defining Worklist Advanced Feed Options

Access the Worklist Advanced Feed Options (WL\_PUB\_AS\_ADVOPT) page (click the Advanced Options link on the Worklist Publish as Feed page).

Worklist

Advanced Feed Options

Specify the advanced options of this feed.

Feed:

Worklist Notification

Feed Options

Max Number of Entries:

10

(Enter 0 for unlimited number of entries.)

Worklist Feed Parameters

Parameter	Value	Description		
Business Process Name	Administer Workflow	Administer Workflow	+	-
Activity Name	Send Note	Send Note	+	-
Event Name	Worklist Note	Worklist Note	+	-
Worklist Name	Worklist Note	Worklist Note	+	-
Status	0	0-Available	+	-
Status	1	1-Selected	+	-

Worklist Advanced Feed Options page

Use this page to determine which worklists are published as feeds.

**Max Number of Entries** Enter the maximum number of feed entries to display in the feed reader.  
(maximum number of entries)

## Worklist Feed Parameters

Select the parameters and values for the worklists that you want to publish. A status of *1* or *0* will always be included. Worklist feeds will only show worklist items that are either *Available* or *Selected*. 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.

When multiple feed parameters are specified, similar parameters use OR, while distinct parameters use AND in the WHERE clause generated by the SQL fetch.

### Worklist Feed Parameters

This table lists worklist feed parameters and their values and data source property:

Parameter Name	Values	Data Source Property
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	<i>1-High, 2-Medium, or 3-Low</i>	WL_PRIORITY
Status	<i>0-Available or 1-Selected</i>	INSTSTATUS
Timed Out	<i>0-Available or 1-Timed Out</i>	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, if the parameters were defined as shown in this table:

Parameter	Value
Business Process Name	Administer Workflow
Activity Name	Send Note
Event Name	Worklist Note
Worklist Name	Worklist Note

<b>Parameter</b>	<b>Value</b>
Priority	1
Status	1
Status	0
Timed Out	0
From	PSADMIN
From	PTSECADM

Then the WHERE clause for the SQL statement generated for 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 system maps the feed properties to Atom 1.0 data source values:

<b>Feed Property</b>	<b>Maps To</b>
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   "/"   ACTIVITYNAME   "/"   EVENTNAME Created On: INSTAVAILABLEDTM Instance ID: INSTANCEID From: ORIGINATORID
Enclosure	none
FullContent	none
GUID	Worklist URL   " "   INSTAVAILABLEDTM
Published	INSTAVAILABLEDTM
Title	INSTANCEID   "."   BUSPROCNAME   "/"   ACTIVITYNAME   "/"   EVENTNAME
Updated	LASTUPDDTM

---

**Note.** INSTAVAILABLEDTM is the time at which the worklist item was created.

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

---

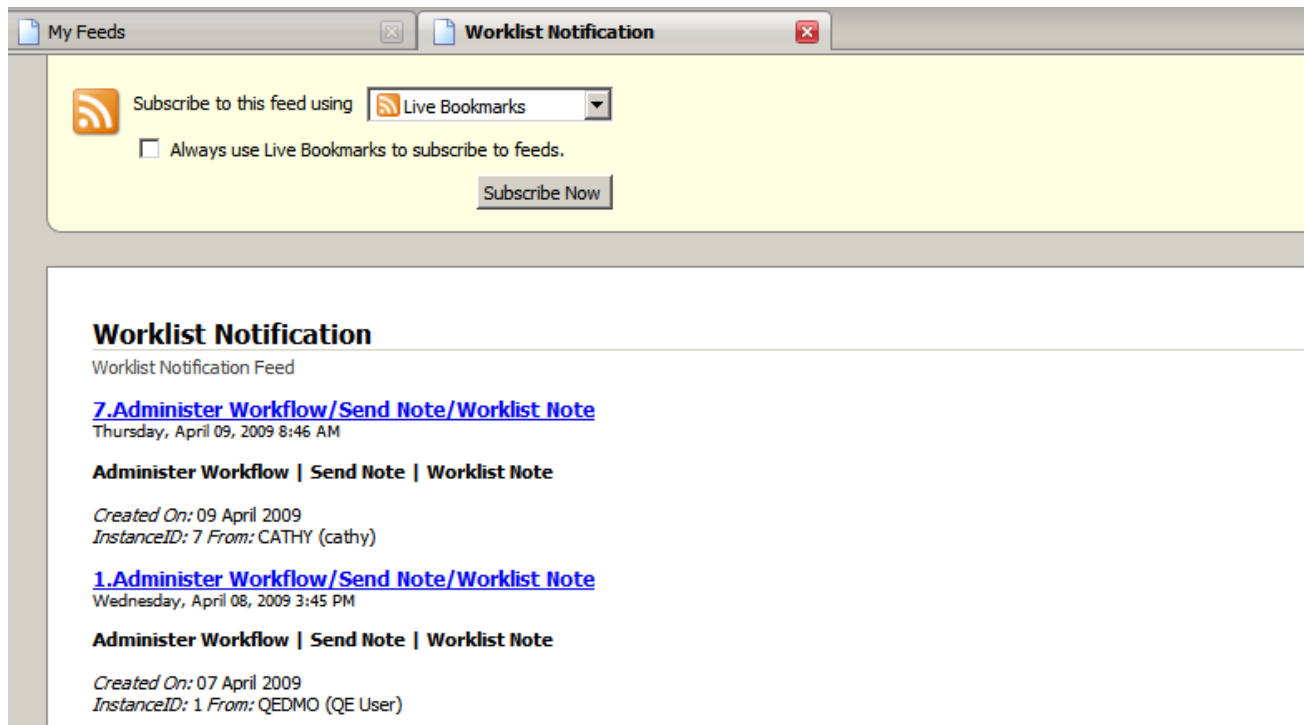
---

## Using Worklist Feeds

Worklist feeds are secured by user ID. The worklist user must have access to the service operation PTFP\_FEED. This access is included in the permission list *PTPT1000*, which is in the role *PeopleSoft User*. When a user is signed on to PeopleSoft Pure Internet Architecture, the user can view the feeds by selecting any of the following navigation paths:

- Select Worklist from the Navigation Bar.  
Click the Feeds link and select the desired worklist feed.
- Select Worklist, Worklist.  
Click the Feeds link and select the desired worklist feed.
- Select Worklist, Worklist Details.  
Click the Feeds link and select the desired worklist feed.
- Select My Feeds.  
Enter search criteria and click Search.  
Select the desired worklist feed.

This example shows a Worklist Notification on the Feeds page:



Example of Feeds page displaying the selected worklist feed

When you click the Worklist item, the appropriate page will appear.

You can subscribe to the feed and select the desired feed reader to use.

When you select a worklist item from a feed reader outside of the PeopleSoft system, the PeopleSoft sign-on screen will appear. Sign into the PeopleSoft system to view and work the worklist item.

## Managing Worklist Feeds

Access the Publish as Feed (PTFP\_PUB\_AS\_LIST) page (click the Publish as Feed link from the Worklist page).

### Worklist

#### Publish as Feed

Review, edit or add feed definitions for this item. Only feed definitions published in the current site are marked as published and can be edited.

Feed Definitions				
		First	1-2 of 2	Last
Feed Title	Published			
1 <a href="#">Order Approval</a>	<input checked="" type="checkbox"/>	Edit	Delete	
2 <a href="#">Worklist Notification</a>	<input checked="" type="checkbox"/>	Edit	Delete	

[Return](#)

[Add Feed](#)

Worklist Publish as Feed page

All of the published worklist feed definitions will appear. From this page, you can edit, delete, or add worklist feeds.

<b>Edit</b>	Use this button to open the Publish Feed Definition page, where you can edit the current published definition.
<b>Delete</b>	Use this button to delete the worklist feed.
<b>Add Feed</b>	Use this button to define and publish a new worklist feed.
<b>Return</b>	Returns the user to the previous page.

## Chapter 8

# Developing New Feed Data Types

This chapter discusses how to:

- 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.
- Develop up-front scheduled feeds.

---

## 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.

---

## Creating the Feed Data Source Application Class

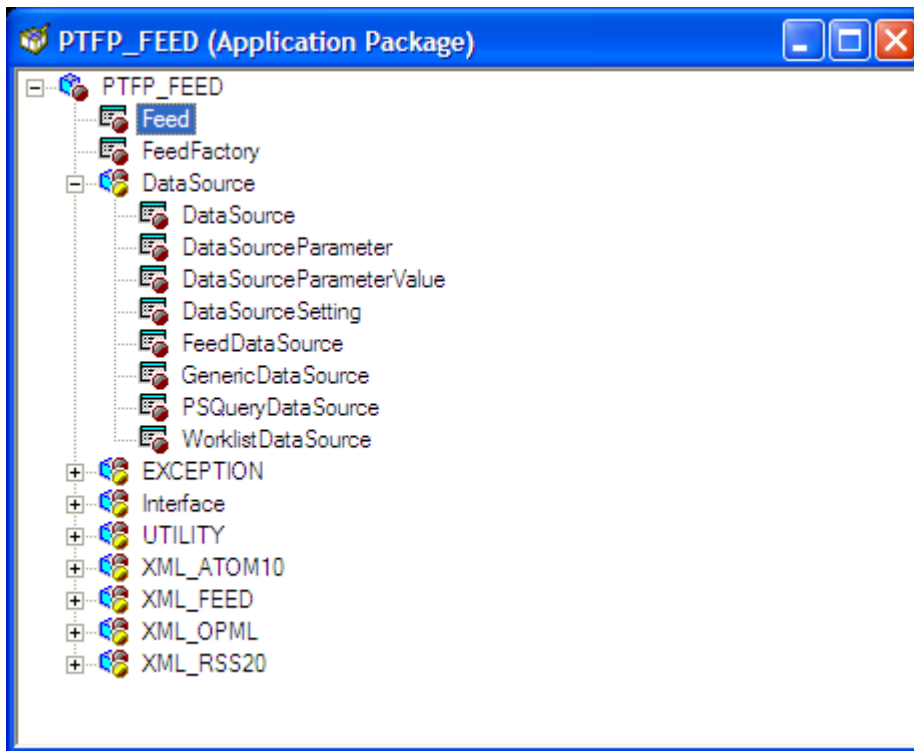
The feed data source application class manages all aspects of data collection and data security.

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 *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Creating Application Packages and Classes."

## Implementing the Methods

Consider these method types:

- Required
- Recommended
- Optional

### **Required Methods**

This table describes the methods that you must implement:

<b>Method</b>	<b>Purpose</b>
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:

<b>Method</b>	<b>Purpose</b>
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:

<b>Method</b>	<b>Purpose</b>
onSave	Perform tasks after saving the feed definition.

<i>Method</i>	<i>Purpose</i>
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 feed data types and discusses how to:

- Define feed data types
- Define advanced options.

## Page Used to Define Feed Data Types

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Define Feed Data Types	PTFP_DATATYPE	PeopleTools, Feeds, Define Feed Data Types	Define feed data types.

## Steps for Defining Feed Data Types

You complete these steps to define feed data types:

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 Feed List feed, which lists all feeds of this data type accessible by the user.

## Defining Feed Data Types

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

### Define Feed Data Types

Define the data types used by the Feed Publishing Wizard



Data Type: PSQUERY

\*Description: PS Query Feed ☒ Active

Long Description: Feeds of this data type contain data of the specified PS Query.

#### Default Feed Properties

Copyright:

Logo:

Icon:

Author Name:

Author Email:

Contributors		Customize	Find	First	1 of 1	Last
	Contributor Name	Contributor Email				
1	<input type="text"/>	<input type="text"/>				

#### Supporting Application Class

Package Name: PTFP\_FEED

Path: DataSource

Application Class ID: PSQueryDataSource

#### Feed Service Operations

		Customize	Find	First	1-2 of 2	Last
	*Service Operation	Description	Type	Default		
1	PT_DYNAMIC_QU	RealtimeQueryFeed	Real Time	<input checked="" type="checkbox"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
2	PT_PREPUBLISH	ScheduledQueryFeed	Scheduled	<input type="checkbox"/>	<input type="button" value="+"/>	<input type="button" value="-"/>

[Publish as Feed](#)

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.

<b>Data Type</b>	The type of feed that you are creating or editing. This field is display only.
<b>Description</b>	Enter a short description of the type of feed. You may enter up to 30 characters.
<b>Long Description</b>	Enter a long description of the type of feed to clearly clarify its purpose. You may enter up to 255 characters.
<b>Active</b>	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.

---

<b>Copyright</b>	Enter copyright information to be included in the XML.
<b>Logo</b>	Enter a URL to the logo to be included in the XML, for example, <i>http://myserver.com/img/logo.gif.</i>
<b>Icon</b>	Enter a URL to an icon to be included in the XML, for example, <i>http://myserver.com/img/icon.gif.</i>
<b>Author Name</b>	Enter an author to be included in the XML.
<b>Author Email</b>	Enter an author email address to be included in the XML.
<b>Contributor Name</b>	Enter a contributor to be included in the XML.
<b>Contributor Email</b>	Enter a contributor email address to be included in the XML.

### ***Supporting Application Class***

<b>Package Name</b>	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.
<b>Path</b>	Enter the application class path that you want to use for the data type.
<b>Application Class ID</b>	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.

**Feed Service Operations**

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

**Defining Advanced Options**

You can create your own Advanced Options page to manage options unique to the new feed data type.

The standard advanced option page, which is used by IB Generic message feeds, has one advanced option. You can control the maximum number of entries to display in the feed reader to the user.

Access the Define Generic Feeds - Advanced Feed Options page.

Define Generic Feeds

Advanced Feed Options

Specify the advanced options of this feed.

Feed Title: DELETE\_ROLE

Feed Options

Max Number of Entries:

(Enter 0 for unlimited number of entries.)

Define Generic Feeds - Advanced Feed Options page

**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.
- 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:

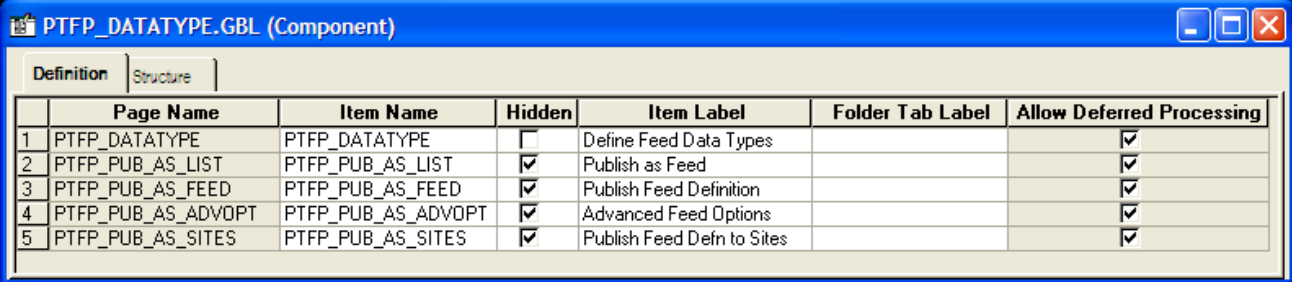
- Publish Feed Definition (PTFP\_PUB\_AS\_FEED)
- Advanced Feed Options (PTFP\_PUB\_AS\_ADVOPT)
- Publish as Feed (PTFP\_PUB\_AS\_LIST)
- Publish Feed Definition to Sites (PTFP\_PUB\_AS\_SITES)

---

**Note.** All four pages can be 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	<input type="checkbox"/>	Define Feed Data Types		<input checked="" type="checkbox"/>
2	PTFP_PUB_AS_LIST	PTFP_PUB_AS_LIST	<input checked="" type="checkbox"/>	Publish as Feed		<input checked="" type="checkbox"/>
3	PTFP_PUB_AS_FEED	PTFP_PUB_AS_FEED	<input checked="" type="checkbox"/>	Publish Feed Definition		<input checked="" type="checkbox"/>
4	PTFP_PUB_AS_ADVOPT	PTFP_PUB_AS_ADVOPT	<input checked="" type="checkbox"/>	Advanced Feed Options		<input checked="" type="checkbox"/>
5	PTFP_PUB_AS_SITES	PTFP_PUB_AS_SITES	<input checked="" type="checkbox"/>	Publish Feed Defn to Sites		<input checked="" type="checkbox"/>

Example of PTFP\_DATATYPE component showing four hidden standard Publish as Feed pages

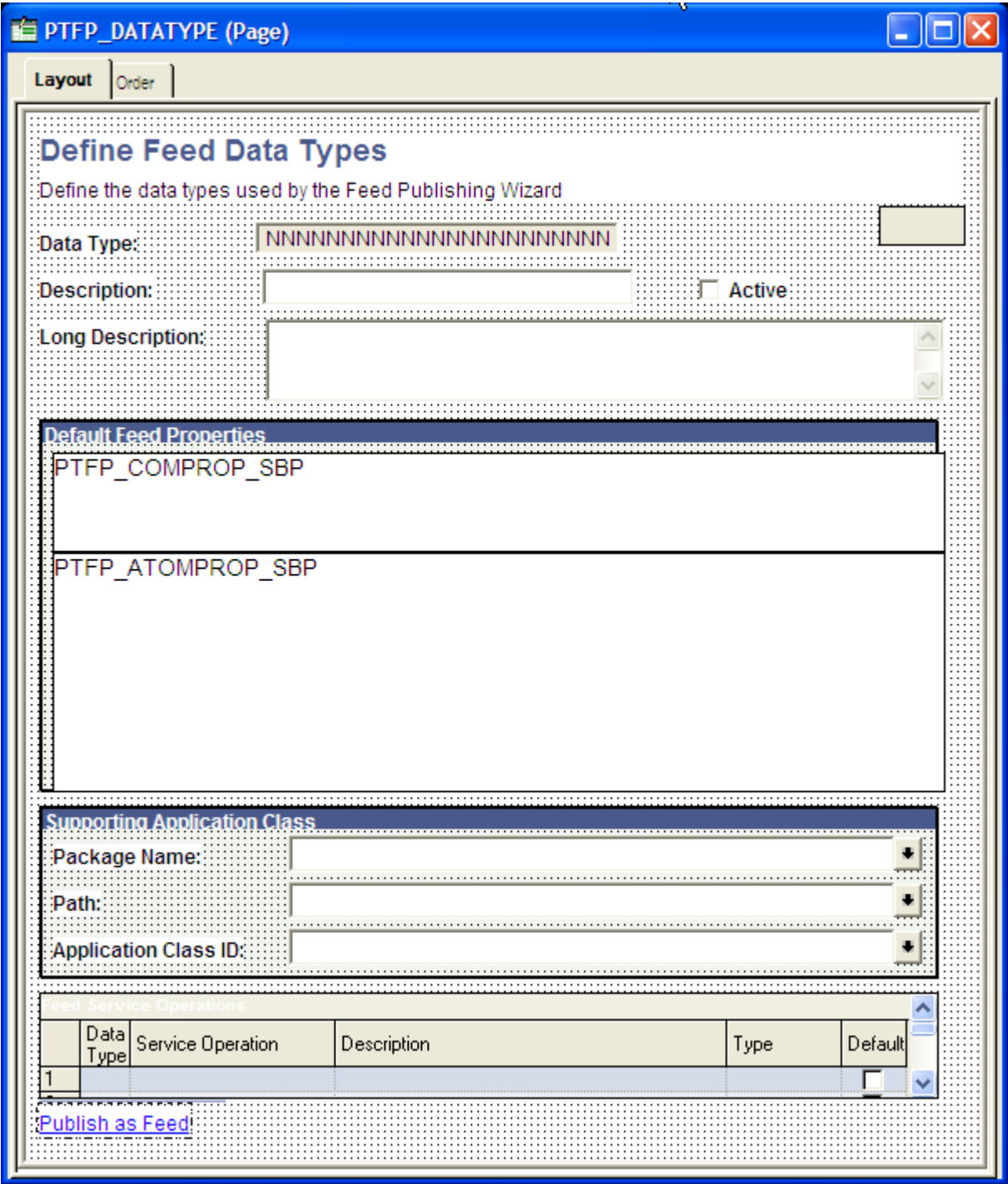
See *Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Developer's Guide*, "Creating Component Definitions," Adding Pages to Components.

## 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:

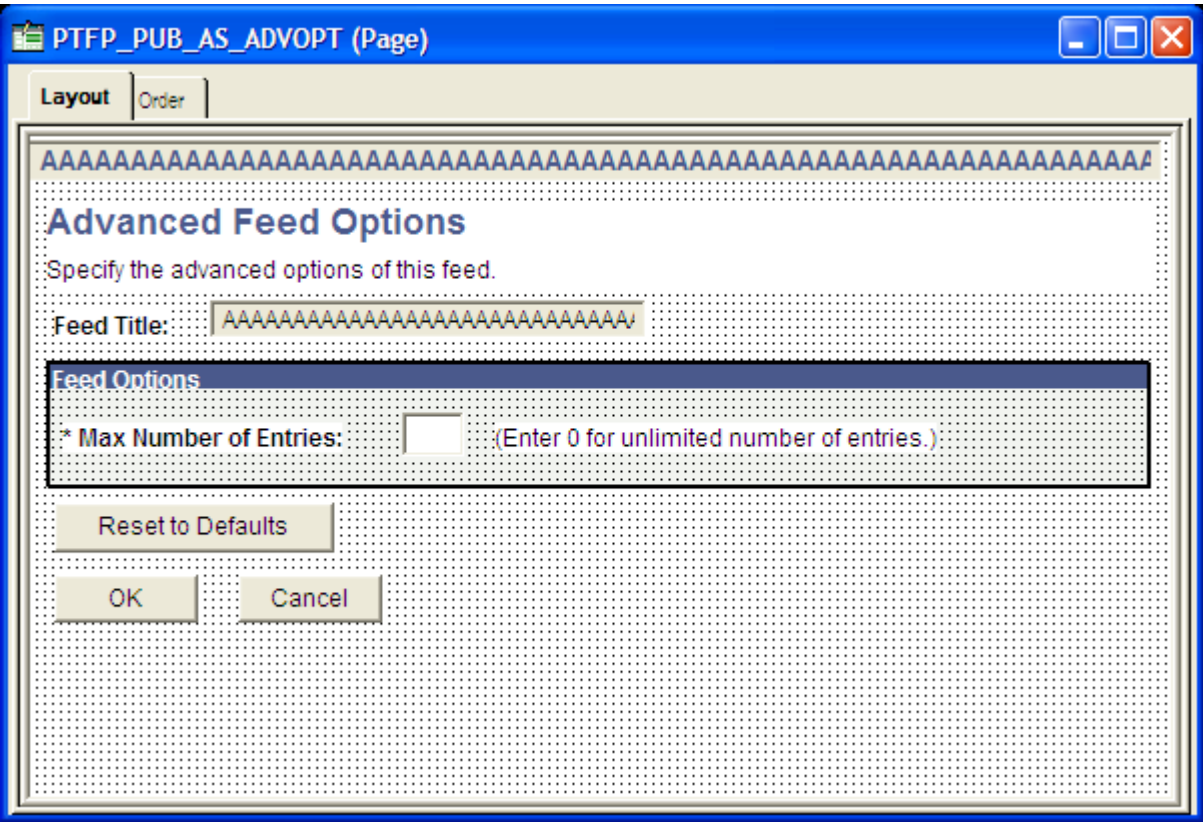


PTFP\_DATATYPE page showing the Publish as Feed link

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Developer's Guide*, "Using Page Controls," Using Push Buttons and Links.

## Creating an Advanced Options Page

If necessary, replace the following standard Advanced Feed Options page (PTFP\_PUB\_AS\_ADVOPT) with a feed data type specific Advanced Options page to configure any properties specific to the new feed data type:



PTFP\_PUB\_AS\_ADVOPT standard Advanced Options page

### Example

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

The screenshot shows a web browser window titled "PTSPQRY\_ADV\_OPT (Page)". The main content area displays the "Advanced Feed Options" dialog box. At the top, there are tabs for "Layout" and "Order". Below the tabs, a header bar contains a long string of "A"s. The main title "Advanced Feed Options" is followed by the instruction "Specify the advanced options of this feed.".

The "Feed Title:" field is filled with a long string of "A"s. Below it, the "Query Prompts" section contains a table with three columns: "Parameter ID", "Description", and "Value". The first row has "1" in the "Parameter ID" column. Below this table is the "Advanced Query Feed Options" section, which includes a "Max Number of Entries:" field (set to 0), a "Preview Feed" link, and three groups of radio button options: "Entry Occurrence" (All Rows in One Feed Entry, One Row Per Feed Entry), "Feed Publishing Type" (Scheduled, Real Time), and "Publish Language" (Current Language Only, All Available Languages). A "Query Feed Security:" dropdown menu is also present.

The "Query Fields" section contains a table with two columns: "Record.Fieldname" and "Heading Text". The first row has "1" in the "Record.Fieldname" column. Below this table is the "Feed Entry Content Mapping" section, which contains a table with three columns: "Comment", "Feed Entry Element", and "Entry Template". The first row has "1" in the "Comment" column. At the bottom of the dialog are five buttons: "OK", "Cancel", "Save", "Save As...", and "Reset".

Query feeds Advanced Option page

## See Also

*Enterprise PeopleTools 8.50 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:

```

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";

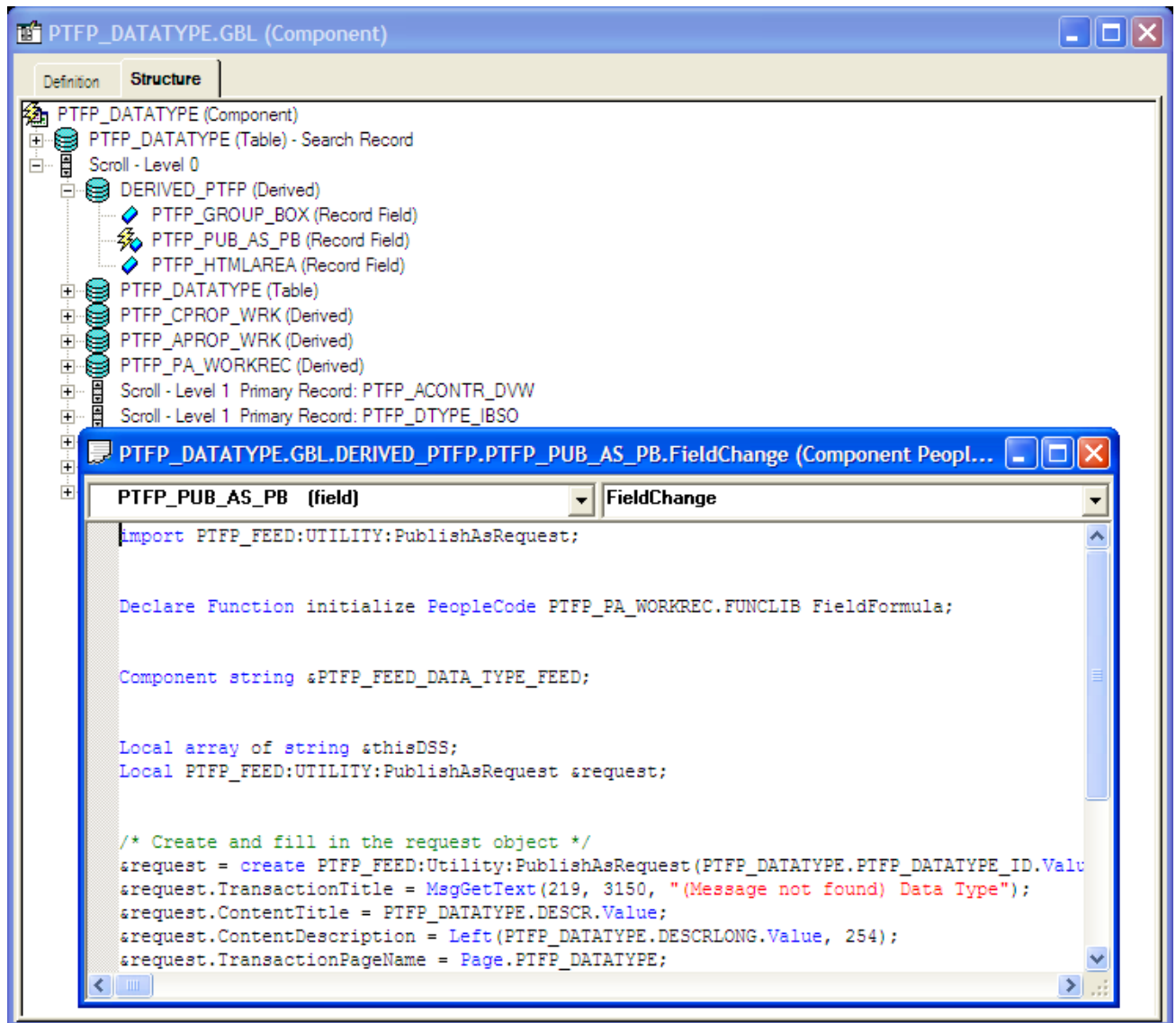
/* Fill in the data source setting values */
&thisDSS = CreateArray("<Data Source Setting Name>", "<Data Source Setting→
Value>");
&request.DataSourceSettings.Push(&thisDSS);

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

```

### **Example**

Notice the Component Record Field PeopleCode on the PTFP\_PUB\_AS\_PB field in this example:



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

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Accessing PeopleCode and Events," Accessing Record Field PeopleCode

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Accessing PeopleCode and Events," Accessing Component Record Field PeopleCode

## Updating the View Content Component or Pagelet

This section discusses how to:

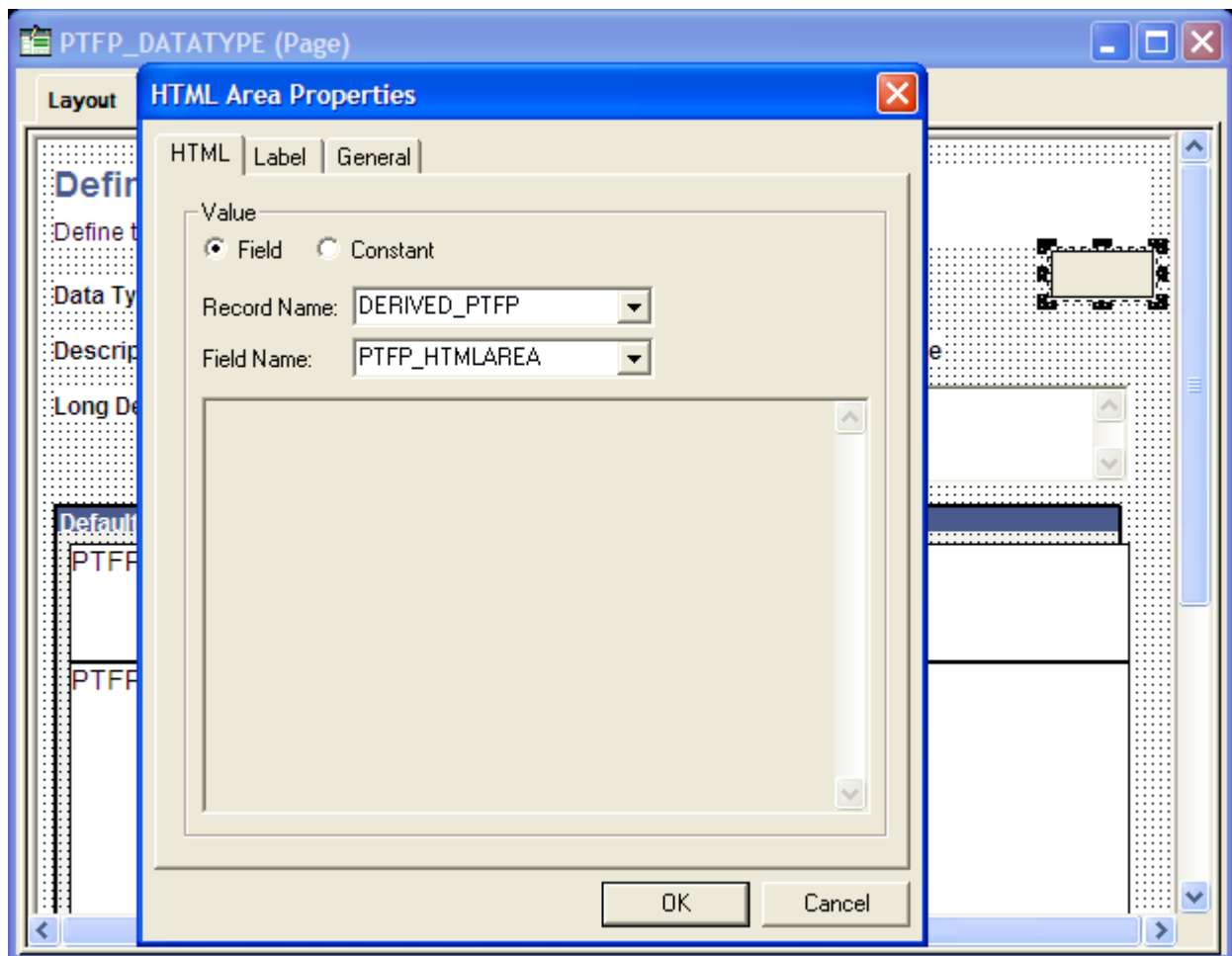
- Add the Related Feeds hover menu to pages.
- Add component or page PeopleCode.
- Add the Related Feeds hover menu to Pagelet Wizard pagelets.

### 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

## See Also

*Enterprise PeopleTools 8.50 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();

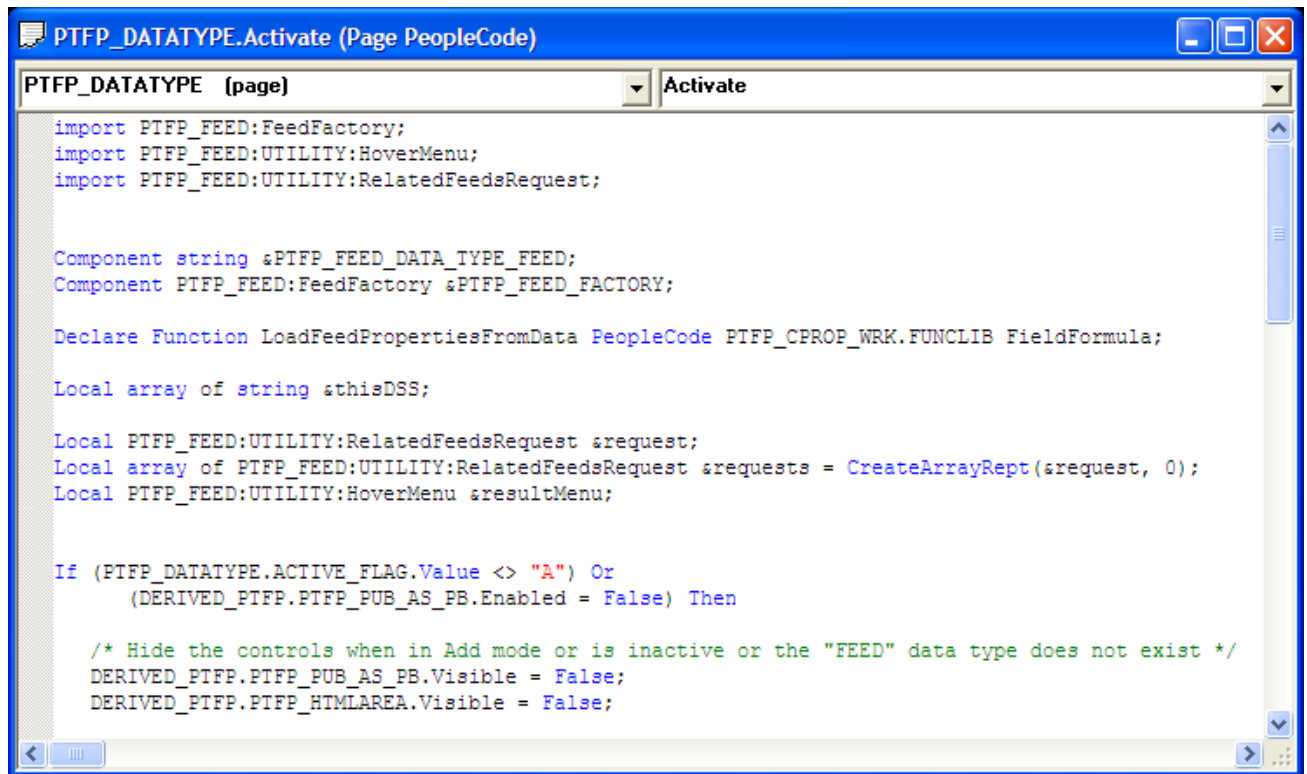
/* Fill in the search criteria */
&request = create PTFP_FEED:UTILITY:RelatedFeedsRequest("<Unique ID>");
&request.DataTypeID = "<yourDataTypeID>";
&request.DataSourceSettings.Push(CreateArray("<Data Source Setting Name>", "<Data⇒
Source Setting Value>"));
&requests.Push(&request);

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 *Enterprise PeopleTools 8.50 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:



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:

```

<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>

```

---

## Developing Up-front Scheduled Feeds

Some additional steps are required for creating upfront scheduled feeds. This section discusses the configuration options:

### Up-front Scheduled Feeds Prerequisites

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

1. The service operation that you use to publish the up-front feed messages to the Integration Broker (IB) 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 IB queue.
2. The service operation used to publish the upfront feed messages has to be listed in the Define Feed Data Types page. This information is used for archiving feeds.

3. The feed format should be set as the message attribute before publishing the message.

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

4. DSPARAMETER\_MAXROW, DSPARAMETER\_SF\_PAGING and DSPARAMETER\_SF\_MAXMINUTES data source parameters found in PTFP\_FEED:UTILITY:Utility application class, should be defined and set to appropriate values. DSPARAMETER\_MAXROW and DSPARAMETER\_SF\_MAXMINUTES are required for archiving feeds. DSPARAMETER\_SF\_PAGING is for paged feeds. It should be set to SF\_PAGINGOPTION\_NOPAGING value. SF\_PAGINGOPTION\_SEGMENTED value is not supported for this release and will return an empty feed.

```
&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_SF_MAXMINUTES, String=>
(&utility.SF_MAXMINUTES_ALLMSG));
&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_ALLMSG);
&thisDSP.Value = &thisDSP.DefaultValue;
&thisDSP.Required = True;

(XRef - Archiving Feeds section at the end of this section)
```

## Appendix A

# Troubleshooting

This appendix provides troubleshooting tips.

---

## Common Problems

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

<b><i>Problem</i></b>	<b><i>Resolution</i></b>
Feed page displays the feed title but no items.	Feed items appear in the context of the signed on user. One of the following would cause a blank list: <ul style="list-style-type: none"><li>• The current user has no feed items.</li><li>• The feed definition was not defined correctly.</li></ul>
Click a feed item and receive this message: Site name is not valid. Check your syntax and try again.	The Portal Context URI text or Portal URI text for the default local node is not configured properly.
Receive this error message when you try to access Feeds: Address not found.	Secure Target Location or Target Location is not set in IB Service Configuration page.
You are signed on to a PeopleSoft Pure Internet Architecture page, and an Authentication Required dialog box appears, enabling you to sign on when you select a worklist item from the feed list.	You used an IP address in the URL to access PeopleSoft and a fully qualified URL in the service configuration for the machine or vice versa.
Receive a message to sign on to the WebLogic server.	For WebLogic servers, you need to configure the server for basic authentication.
User is challenged for authentication after signing in when requesting feeds.	Use HTTP over SSL since basic authorization is not secure. Feeds should use secure target location when available.  Mismatch in protocol can cause this. If Feed URL is HTTP (that is, Target Location was specified) but the PIA was accessed using HTTPS and vice versa.



## Appendix B

# Configuring Integration Broker to Work With the Feed Publishing Framework

When using the Feed Publishing Framework, you must configure Integration Broker. This appendix discusses how to:

- Configure the IB gateway and node application server.
- Configure the service target location.
- Configure the default user of the ANONYMOUS node.
- Configure the content URI and portal URI node properties.

---

## Configuring the IB Gateway and Node Application Server

To configure the IB gateway and node application server:

1. Select PeopleTools, Integration Broker, Configuration, Quick Configuration.
2. Enter the machine name in the Gateway URL field.
3. Click the Ping Gateway button.

The status should return as *Active*.

4. Click the Advanced Gateway Setup link.

This opens the Gateways page.

5. Click the Gateway Setup Properties link and login.

This opens the PeopleSoft Node Configuration page.

6. Enter the appropriate values in the Gateway Default App. Server fields.
7. Enter the appropriate values in the PeopleSoft Nodes fields.
8. Click the Ping Node button.

The status should return as *Success*.

9. Click the Return button.

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

This opens the Gateway Properties page.

11. If the web server goes through a proxy server to connect to the internet, enter the proxy information.
12. Verify that the secureFileKeystorePasswd is correct and encrypted.
13. Click the OK button.
14. On the PeopleSoft Node Configuration page, click the Save button.

---

## Configuring the Service Target Location

To configure the service target location:

1. Select PeopleTools, Integration Broker, Configuration, Service Configuration.
2. Enter the Target Location value to point to the appropriate IB Gateway.
3. Save the page.

---

## 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. Open the ANONYMOUS node.
3. Select the Node Definitions tab and verify that the Default User Id field contains a valid user with limited privileges.

---

## Configuring the Content URI and Portal URI Node Properties

To configure the content URI and portal URI node properties:

1. Select PeopleTools, Integration Broker, Integration Setup, Nodes.
2. Open the default local node.
3. Select the Node Definitions tab and verify that the Authentication Option field value is *Password* or *Certificate*.
4. Select the Portal tab and verify that the Content URI Text and the Portal URI Text fields point to the correct database.
5. Verify that the node type is *PIA*.

6. Save the component.



## Appendix C

# Configuring Oracle WebLogic Servers for Basic Authentication

When using the Feed Publishing Framework, you must configure Oracle WebLogic servers for basic authentication. This appendix discusses how to modify the config.xml file.

---

## Modifying the config.xml File

Perform the following tasks:

1. Stop the web server.
2. Access the PS\_HOME\webserv\WEBSERVER\_DOMAINNAME\config folder.
3. Open the config.xml file.

4. Add the tag `<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>` in between the `<security-configuration>` tags. For example:

```
<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/ffmyP16a=>
D4NKEv0va5IxytcWvGRV50mB5dYbzhos9XWNN0Lz4mQoXiy</credential-encrypted>
  <node-manager-username>system</node-manager-username>
  <node-manager-password-encrypted>{3DES}c/WblCLbZubUdNamvjN1sw==</node-manager->
password-encrypted>
```

---

**Note.** The modification is the next line.

---

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

</security-configuration>
```

5. Save the file and start the web server.

---

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

---

# Index

## A

- Advanced Feed Options page 22, 41, 49, 82
- Advanced Feed Options page, creating 82
- advanced options, defining 79
- application class
  - feed data source, creating 74
  - supporting 78
- application classes
  - design time role 10
  - run time role 10
- archiving
  - feeds 35
  - Integration Broker generic message feeds 35
  - options and logic 37
  - parameters 36

## B

- base class, extending 74
- bind variables, accessing in rich text editor 60

## C

- common problems, troubleshooting 91
- component or page PeopleCode, adding 87
- creating feeds 17

## D

- data feed types, steps to defining 76
- data-level, security 14
- data types
  - analyzing requirements for new 73
  - developing 73
  - FEED 11
  - GENERICFEED 11
  - PSQUERY 11
  - WORKLIST 11
- Define Feed Data Types page 45, 77
- Define IB Generic Message Feeds page 40
- document properties, element mapping 8

## E

- entry templates
  - assembling elements 56
  - creating in rich text editor 60
  - editing with Mapping Builder 56
- exporting, feed definitions 34

## F

- feed
  - properties 21
  - properties, additional 21
  - security options 21
  - security options, selected 22
  - service operations 79
- FEED data type 11
- feed definitions
  - exporting 34
  - importing 34
- Feed Element Mapping Builder page 57
- feed entry
  - content mapping 52
  - mapping elements to entry templates 51
  - table of element values 52
- feed-level security 13
- feed list feeds, publishing 24
- feed properties 66
  - default 78
  - defining 19
- Feed Publishing Framework
  - architecture 5
  - creating a new type of feeds 10
  - data type application classes 9
  - diagram of architecture 6
  - documentation generation and delivery 7
  - document properties 8
  - feed-level security 13
  - implementation 2
  - overview 1
  - prerequisites 2
  - runtime engine 6
- feed publishing pages
  - Advanced Feed Options 22
  - Publish Feed Definition 19
  - Publish Feed Definition to Sites 23
- feeds
  - advanced options 22
  - archiving 35
  - archiving options and logic 37
  - archiving parameters 36
  - creating and using 17
  - defining properties 19
  - developing new data types 73
  - developing upfront scheduled 89
  - diagram of process flow 18
  - managing 33
  - migrating between databases 33
  - process flow 18
  - publishing 14, 19
  - publishing to additional sites 23
  - real-time 12
  - scheduled 12
  - security to create data types 13
  - security to publish 13
  - security to view 13
  - viewing 30

## G

GENERICFEED data type 11  
GetFeed, service operation 7

## I

implementing methods  
    optional 75  
    recommended 75  
    required 75  
importing, feed definitions 34  
Integration Broker  
    deleting generic message feeds 42  
    editing generic message feeds 42  
    publishing generic message feeds 40  
    understanding generic message feeds 39

## M

managing feeds 33  
Mapping Builder, use to edit entry templates 56  
methods  
    implementing 74  
    protected, setting flags 76  
migrating feeds between databases 33  
moving, seed data 33  
My Feeds, searching for user specific feeds 14  
My Feeds page 25

## N

new feed data types  
    analyzing requirements 73  
    steps for developing 4

## O

opml files, generating 28  
Oracle WebLogic Server  
    configuring for authentication 97  
    modifying the config.xml file 97

## P

pages  
    PTFP\_DATATYPE 45, 63, 77  
    PTFP\_GENERIC\_FEED 40  
    PTFP\_PUB\_AS\_ADVOP 22, 41, 49, 82  
    PTFP\_PUB\_AS\_FEED 19, 62, 65  
    PTFP\_PUB\_AS\_LIST 42, 71  
    PTFP\_PUB\_AS\_SITES 23  
    PTFP\_VIEW 25  
    WL\_PUB\_AS\_ADVOP 67  
PeopleTools-delivered feed data types, using 3  
properties

    feed 21  
    feed, additional 21  
property maintenance component, updating 79  
PSQUERY data type 11  
PTEP\_GETFEED, service operation 7  
PTFP\_GETFEEDLIST 28  
PTIBFEEDARCH process 36  
Publish as Feed link 80  
Publish as Feed page 62, 65, 71  
Publish as Feed page, adding link to a page 80  
Publish Feed Definition page 19  
Publish Feed Definition to Sites page 23  
publishing  
    feed list feeds 24  
    feeds 14  
    to additional sites 23  
    worklist feeds 65

## Q

Query feeds  
    advanced options 50  
    data types 45  
    defining advanced feed options 48  
    managing 48, 62  
    publishing and managing definitions 47  
    security 46  
    steps to publishing 47  
    understanding 45  
    using 61

## R

record PeopleCode, adding 83  
related feeds 27  
related feeds hover menu 27  
Related Feeds hover menu  
    adding to Pagelet Wizard pagelets 88  
    adding to pages 86  
Run Feed Archive page 35  
runtime engine, process flow 7

## S

security  
    data-level 14  
    feed-level 13  
    to create feed data types 13  
    to publish feeds 13  
    to view feeds 13  
security options  
    feed 21  
    selected 22  
seed data, moving 33  
service operations  
    feed 79  
    GetFeed (PTEP\_GETFEED) 7  
    PTFP\_GETFEEDLIST 28  
standard publish as feed pages, adding  
    Advanced Feed Options  
        (PTFP\_PUB\_AS\_ADVOP) 80  
    Publish as Feed (PTFP\_PUB\_AS\_LIST) 80

- Publish Feed Definition  
     (PTFP\_PUB\_AS\_FEED) 80
- Publish Feed Definition to Sites  
     (PTFP\_PUB\_AS\_SITES) 80
- SysAudit information
  - for feed definition integrity 35
  - for generic IB message feed integrity 35
  - for PS Query feed integrity 35
  - for worklist feed integrity 35

## T

- troubleshooting, common problems 91

## U

- user specific feeds, My Feeds 14

## V

- viewing feeds 30

## W

- Worklist Advanced Feed Options page 67
- Worklist Data Feed Type page 63
- WORKLIST data type 11
- worklist feed
  - additional configuration 63
  - data types 63
  - understanding 63
- worklist feeds
  - advanced feed options 67
  - defining 65
  - managing 71
  - parameters 68
  - publishing 65
  - steps for using 70
  - steps to publish 65

