ORACLE
PeopleSoft

PeopleTools 8.60: Integration Broker Administration

July 2024

ORACLE

Documentation Accessibility

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

Access to Oracle Support

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

Contents

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

Hosted PeopleSoft Online Help

You can access the hosted PeopleSoft Online Help on the Oracle Help Center. The hosted PeopleSoft Online Help is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support. The hosted PeopleSoft Online Help is available in English only.

To configure the context-sensitive help for your PeopleSoft applications to use the Oracle Help Center, see Configuring Context-Sensitive Help Using the Hosted Online Help Website.

Locally Installed PeopleSoft Online Help

If you're setting up an on-premises PeopleSoft environment, and your organization has firewall restrictions that prevent you from using the hosted PeopleSoft Online Help, you can install the online help locally. Installable PeopleSoft Online Help is made available with selected PeopleSoft Update Images and with PeopleTools releases for on-premises installations, through the Oracle Software Delivery Cloud.

Your installation documentation includes a chapter with instructions for how to install the online help for your business environment, and the documentation zip file may contain a README.txt file with additional installation instructions. See *PeopleSoft 9.2 Application Installation* for your database platform, "Installing PeopleSoft Online Help."

To configure the context-sensitive help for your PeopleSoft applications to use a locally installed online help website, see Configuring Context-Sensitive Help Using a Locally Installed Online Help Website.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format on the Oracle Help Center. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The two main types of common help are:

- Application Fundamentals

- Using PeopleSoft Applications

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft applications.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

Typographical Convention	***Description***
Key+Key	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For **Alt+W,** hold down the **Alt** key while you press the **W** key.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Typographical Convention	*Description*
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region and Industry Identifiers

Information that applies only to a specific region or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in the PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in the PeopleSoft Online Help:

- USF (U.S. Federal)

- E&G (Education and Government)

Translations and Embedded Help

PeopleSoft 9.2 software applications include translated embedded help. With the 9.2 release, PeopleSoft aligns with the other Oracle applications by focusing our translation efforts on embedded help. We are not planning to translate our traditional online help and PeopleBooks documentation. Instead we offer very direct translated help at crucial spots within our application through our embedded help widgets. Additionally, we have a one-to-one mapping of application and help translations, meaning that the software and embedded help translation footprint is identical—something we were never able to accomplish in the past.

Using and Managing the PeopleSoft Online Help

Select About This Help in the left navigation panel on any page in the PeopleSoft Online Help to see information on the following topics:

- Using the PeopleSoft Online Help.
- Managing hosted Online Help.
- Managing locally installed PeopleSoft Online Help.

PeopleTools Related Links

PeopleTools 8.60 Home Page

PeopleSoft Search and Insights Home Page

“PeopleTools Product/Feature PeopleBook Index” (Getting Started with PeopleTools)

PeopleSoft Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

Contact Us

Send your suggestions to psoft-infodev_us@oracle.com.

Please include the applications update image or PeopleTools release that you’re using.

Follow Us

Icon	*Link*
	Watch PeopleSoft on YouTube
	Follow @PeopleSoft_Info on X.
	Read PeopleSoft Blogs
	Connect with PeopleSoft on LinkedIn

Chapter 1

Getting Started with PeopleSoft Integration Broker Administration

PeopleSoft Integration Broker Administration Overview

This subject describes how to perform system administration tasks in PeopleSoft Integration Broker such as:

- Set up and configure integration system components, such as messaging servers, nodes, integration gateways, listening and target connectors, and so on.
- Configure the integration system to handle services, including specifying namespaces, setting up UDDI repositories, and so on.
- Secure the integration environment by applying security at the web server, gateway, application server, node and service operation level.
- Fine tune integration system performance by employing failover, primary/secondary processing, load balancing, and so on.
- And more.

Administering PeopleSoft Integration Broker

PeopleSoft Integration Broker is installed as part of the PeopleTools installation process. Information about configuring the integration gateway, creating service operations and administering integrations is described later in this subject. This section provides information to consider before you begin to use PeopleSoft Integration Broker.

Planning the Integration Architecture

The two major components of PeopleSoft Integration Broker are the integration gateway and the integration engine. The integration gateway is a platform that manages the receipt and delivery of messages passed among systems through PeopleSoft Integration Broker. The integration engine is an application server process that routes messages to and from PeopleSoft applications as well as transform the structure of messages and translates data according to specifications that you define.

Evaluate historical integration data, current data, as well as expected growth and increased traffic. Consider how many interfaces you have in production and how much system resources they use. Also consider how many of these interfaces will remain nightly batch file loads versus how many do you want to be real-time service based integrations. Devise simulated real-life integration scenarios where you can estimate volume and size of the transactions to a certain degree. Then use this information for benchmarking and stress testing, which should lead to performance tuning, hardware sizing, and so on.

Understanding Integrations Processed by the Integration System

Work with development teams to understand the type, number and frequency of integration that will be processed on the system. Doing so will assist you in setting up and configuring components properly, as well as in performance tuning the system.

Consider the following:

- Real-time integrations or scheduled integrations.
- Determine if your business needs require using real-time integration, scheduled integrations, or a combination of both. Scheduled batch processing and file loads of scheduled integration may impact system performance and the running of other system applications.
- Inventory the integration being developed and performed.
- Determine which systems and applications will participate in each integration. Consider dependencies on other systems owned by other groups having concurrent releases, and data dependencies within the context of synchronizing data between systems. Do you need permission from business owners to integrate with their systems?
- Synchronous integrations and asynchronous integrations.

  In PeopleSoft Integration Broker synchronous integrations, all processing stops after the system sends a request to an integration partner, until a response is received back from that partner. In PeopleSoft Integration Broker asynchronous integrations, each request is placed in a queue to be processed as soon as the system can accommodate the request.

  Synchronous integration processing and asynchronous integration processing each place different loads on the integration system. Understanding the processing that takes place on your system can help you better tune the system for optimal performance.

Determining Security

Unlike a public web service on the internet that retrieves a stock quote for a given ticker symbol, the web services and integrations in your PeopleSoft applications can expose sensitive information such as financial data. PeopleSoft Integration Broker facilitates transfer of information between systems; however, a security analyst must evaluate security requirements for each individual integration.

For example, security requirements might differ when interfacing with credit card processing vendors, versus publishing salary information out of human resources, versus synchronizing business units between applications, and so on. Perhaps certain information should be available to the public, including systems outside of your company, such as how many inventory items are available for sale. Other information might be restricted to internal employees only, internal application systems only, or perhaps only certain users of a particular application system.

PeopleSoft Integration Broker allows you to secure each individual integration to the level of security required as well as all integration data flowing over the wire.

Accessing Staff Skills

Administrators of PeopleSoft Integration Broker should have familiarity, training or experience in the following areas:

- PeopleTools.

- Web server administration.
- Application server administration.
- Performance testing and tuning knowledge.

Chapter 2

Understanding Setting Up PeopleSoft Integration Broker

Determining the Messaging Architecture

A key step in creating and implementing integrations is to determine what systems to integrate and the architecture to use. For example, your purpose might be to integrate with other PeopleTools systems where a firewall is involved, integrate with third-party systems, or integrate with PeopleSoft 8.4x systems.

The product documentation for Integration Broker features a topic that provides overview information about several messaging architecture scenarios.

Related Links

"Understanding the Integration Scenarios" (Integration Broker)

Installing PeopleSoft Integration Broker

PeopleSoft Integration Broker components are installed during a PeopleSoft environment installation, which includes:

- PeopleTools
- PeopleSoft application software
- Database for the PeopleSoft application
- Application Server
- Web server (PIA)
- Process Scheduler

The PeopleSoft Integration Broker integration engine is installed during the PeopleTools installation process.

The integration gateway is installed as part of the PeopleSoft Pure Internet Architecture installation process.

See *PeopleTools 9.2 Application Installation* for your database platform and your web server documentation.

Installing Digital Certificates on the Integration Gateway

To successfully integrate with PeopleSoft and third-party integration partners, you must install digital certificates on the machine where the integration gateway is installed.

To configure the integration gateway one of the steps you must complete is to the path to the gateway certificate keystore and encrypt and define the keystore password.

Warning! Integrations will fail if the keystore path and encrypted keystore password are not defined in the integration gateway properties file, integrationGateway.properties.

If the integration gateway is installed on a web server that has SSL/TLS implemented, the integration gateway and web server share the digital certificates. As a result, you do not need to install separate integration gateway certificates. When you configure the integration gateway the system uses the SSL/TLS keystore path and you need only specify the encrypted SSL/TLS keystore password in the integration gateway properties file.

If the integration gateway is installed on a web server that does not have SSL/TLS implemented, you must install gateway-based digital certificates and specify the keystore path and keystore password for the certificates in the integration gateway properties file.

Related Links

Understanding Securing Integration Environments
Understanding Digital Certificates
Installing Integration Gateway-Based Digital Certificates

Configuring and Starting Messaging Servers for Asynchronous Messaging

Before using PeopleSoft Integration Broker for asynchronous integrations, you must configure and start the messaging server using PSADMIN.

See Understanding Messaging Server Administration.

Activating Pub/Sub Server Domains

You must activate the domain on which the pub/sub server resides before you can use the messaging server.

To activate pub/sub server domains, use the Domain Status page in the Integration Broker Service Operations Monitor.

Related Links

Understanding Managing Pub/Sub Domains

Defining Integration Gateways and Loading Connectors

PeopleSoft Integration Broker is delivered with one local gateway, *LOCAL*, defined. You can use this gateway as the default local gateway, or create a new gateway and designate that one as the default local gateway.

After you access the delivered local gateway or create your own, you must specify its URL and save the changes. The gateway URL is typically the following:

```
http://<machine_name>:<port>/PSIGW/PeopleSoftListeningConnector
```

The integration gateway URL is case sensitive.

Next you must click the **Load Gateway Connectors** button to load the connectors delivered with PeopleSoft Integration Broker.

Related Links

Understanding Managing Integration Gateways

Configuring Integration Gateway Properties

After you define the default local integration gateway, specify the integration gateway URL and load the delivered connectors, there are additional required and optional gateway properties to set. You set these properties using the integrationGateway.properties file.

At a minimum you must set the following in the integrationGateway.properties file:

- Set the Oracle Jolt connection string parameters in the DELIVERED CONNECTOR CONFIGURATION Section of the file. In most situations, you set the parameters under “JOLT connect string settings for Application Server(s) with known NODENAMEs.”
- Specify and encrypt the keystore password.

Related Links

Accessing the integrationGateway.properties File
Configuring Security and General Properties

Configuring the Integration System to Handle Services

To create services, service operations and generate WSDL documents, you must configure the system to handle services.

PeopleSoft Integration Broker features a Services Configuration page where you must specify the following items before you can create and work with services: services namespace, schema namespace and target location.

Related Links

Understanding Configuring PeopleSoft Integration Broker for Handling Services

Chapter 3

Administering Messaging Servers for Asynchronous Messaging

Understanding Messaging Server Administration

This section discusses messaging servers, messaging server processes, and dedicated messaging servers.

Messaging Servers

The PeopleSoft messaging infrastructure is the core system upon which PeopleSoft Integration Broker is built. Before using Integration Broker for asynchronous message processing, you must configure and start the messaging server.

Note: The messaging servers and messaging server processes are used for asynchronous integrations only. If you are performing only synchronous integrations, you need not configure a messaging server.

Activating Messaging Server Domains

Pub/sub server domains are delivered inactive, and you must activate them for the pub/sub system to become available.

However, if the domain is in Production mode as defined on the Service Configuration page, then the Integration Broker domain status value is set to Active by default.

See Using the Service Configuration Page to Set Service Configuration Properties.

Use the Domain Status page in the Service Operations Monitor to activate pub/sub server domains.

See Understanding Managing Pub/Sub Domains.

Messaging Servers in the Db2 for z/OS Environments

For Db2 for z/OS environments, PeopleSoft delivers messaging servers with persistent cursors off. Therefore, all SQL statements are compiled each time they are invoked.

To change the persistent cursors setting:

1. In PSADMIN locate the Values for config section — Publish&Subscribe.
2. Set the Persistent Cursors on DB2/OS390 option. The values are:
    - *0*: Persistent cursors off.

- *1*: Persistent cursors on.

Messaging Server Processes

Although the server processes devoted to the messaging system are all part of the larger application server domain, they comprise a distinct set of processes that aren't involved with the ordinary transactions associated with PeopleSoft Pure Internet Architecture connections.

Six processes of two types—dispatchers and handlers—are paired to produce the messaging servers that transmit asynchronous messages throughout the messaging system. A set of three messaging servers—a publication broker, a publication contractor, and a subscription contractor—is required by PeopleSoft Integration Broker. The following table lists the generic names for the processes:

Messaging Server	***Dispatcher Name***	***Handler Name***
Publication Broker (BRK)	PSBRKDSP	PSBRKHND
Publication Contractor (PUB)	PSPUBDSP	PSPUBHND
Subscription Contractor (SUB)	PSSUBDSP	PSSUBHND

To distinguish the messaging servers, the PeopleSoft Server Administration utility (PSADMIN) includes a separate menu for administering them—the Messaging Server Administration menu. You select this menu from the PeopleSoft Domain Administration menu.

This example illustrates the PeopleSoft Domain Administration menu. Choose command *7* (Messaging Server Administration menu) to administer messaging servers.

```
--------------------------------
PeopleSoft Domain Administration
--------------------------------
     Domain Name: TEST_QEDMO

  1) Boot this domain
  2) Domain shutdown menu
  3) Domain status menu
  4) Configure this domain
  5) TUXEDO command line (tmadmin)
  6) Edit configuration/log files menu
  7) Messaging Server Administration menu
  8) Purge Cache
  9) Preload File Cache
 10) Clean IPC resources of this domain
  q) Quit

Command to execute (1-10, q) :
```

From this menu, you can create new messaging servers, edit the queue list for existing messaging servers, and delete messaging servers that are no longer needed.

Note: Although you add new messaging servers using a separate menu, you configure the messaging server processes with PSADMIN as you would any other server process.

Related Links

"Understanding PSADMIN Menus" (System and Server Administration)

Understanding Dedicated Messaging Servers

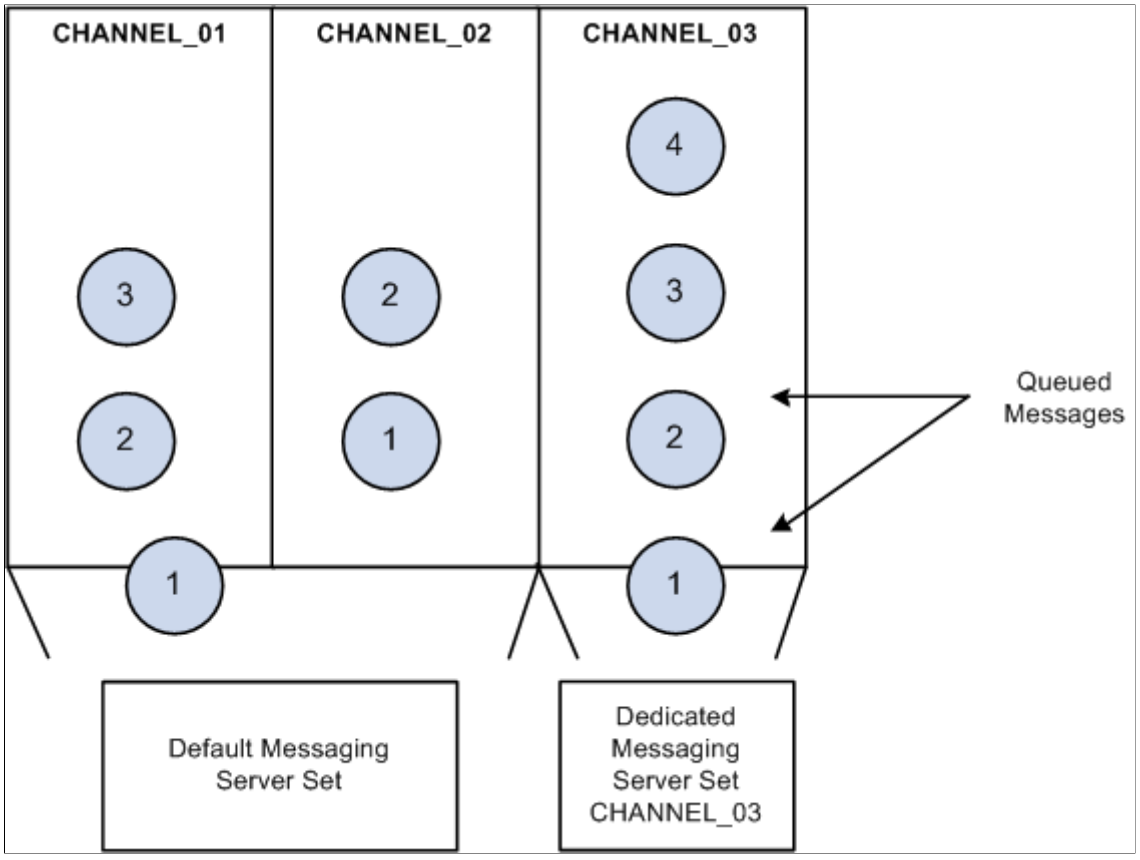
When you create a new application server domain, PSADMIN offers a set of messaging server processes that comprise the default messaging server set for that domain. The default messaging server set is sufficient for development, testing, or demonstrations.

You might use the default messaging server set as the only messaging server set; however, in most cases, it is insufficient. As the volume of published messages increases in a production system, it's likely that a single messaging server set will become overloaded. To avoid potential overloads and performance degradation, create additional dedicated messaging servers to cope with an increase in message volume.

Note: Dedicated messaging servers are used only for asynchronous messaging.

When you create a new messaging server, you assign it to a particular queue using PSADMIN. If a given queue is the most active and creates performance bottlenecks, you can dedicate several messaging servers to that queue to cope with the message volume. A messaging server is capable of handling multiple message queues.

The following illustration depicts a dedicated messaging server set assigned to *QUEUE_03*.



In this scenario, the default messaging server set (_dflt process collection) continues to process the messages in the other message queues while the dedicated messaging server set processes only the messages within a specified queue. Unless you create and configure dedicated messaging servers, the

default messaging server set handles all incoming messages. Remember that a messaging server set is a collection of six messaging server processes.

Note: Before you can assign messaging servers to message queues, you must first define the message queues using PeopleSoft Application Designer.

The process for adding a dedicated messaging server includes two parts:

- Creating the new messaging server.

  Use the Messaging Server Administration menu in PSADMIN. This is where you specify the type of server you're adding, name the server, and assign it to specific message queues.

- Configuring the new messaging server.

  When you add a new messaging server of any type, the configuration files are updated to include parameters for the new server processes. Because a messaging server consists of two server processes, when you create a new one, you'll see two additional configuration sections in the PSADMIN domain configuration menu. They appear identical to the _dflt messaging server processes, except they have the name that you gave them in place of the *_dflt*. For any new messaging server processes to take effect, you must first reconfigure the domain to include the new parameters.

Note: Typically, you add multiple messaging elements simultaneously, so you should create all the elements and then reconfigure the domain once.

Considerations When Creating Dedicated Servers

When creating dedicated messaging servers, consider the following points:

- There is no validation checking when you enter service operation queue names in PSADMIN. As a result, if service operation queue names are not spelled correctly and match those defined in the system, the dedicated server will not process any service operation. Instead the default server will process them.

- Never split a service operation queue across domains. You don't want a situation where a service operation queue is assigned to Domain A and the same service operation queue is also assigned to Domain B, since both domains will try to do the same work. You want specific service operation queues for Domain A and specific service operation queues for Domain B.

- Setting up a dedicated server consists of a creating a dedicated dispatcher and handler(s). Make sure that the number of handlers booted is sufficient to process the request volume.

- If you create more than one dedicated server over different domains do not to include any service operation queues already specified for other dedicated servers of the same server type. For example, do not include Service Operation Queue A in Publication Broker Server X, as well as in Publication Broker Server Y.

- Verify that the Oracle tuxedo queue size is large enough and correctly configured in PSADMIN.

  See Setting the Oracle Tuxedo Queue Size.

- If you choose to set up group domain failover for dedicated servers, ensure that:

- Service operation queue sets within groups are identical.
- Service operation queue sets between groups are unique.

  See Setting Up Domain Failover.

- When you create a messaging server, the following dispatcher parameters are populated with their default values. Verify those default settings you want to keep and those that you want to change.
  - Restart period.
  - Scan interval.
  - Dispatcher queue maximum queue size.
  - Memory queue refresh rate.

    See Specifying Dispatcher Parameters.

Creating and Assigning Dedicated Servers

Typically, you create one server of each type to produce a complete messaging server set dedicated to one or more service operation queues.

Note: Although a messaging server set consists of one of each of the three server types, they do not all need to be dedicated servers. For example, for a given service operation queue, you can create only a dedicated publication contractor. If you haven't assigned a dedicated publication broker or a dedicated subscription contractor to the service operation queue, the default publication broker and subscription contractor is used.

This example illustrates the Messaging Server Administration menu that you use to create a new messaging server.

```
------------------------------------------------------------
              Messaging Server Administration menu
------------------------------------------------------------
  Domain Name : QEDMODW

  In addition to the default messaging servers, the following
  dedicated messaging servers are in the domain configuration:

       << No dedicated messaging servers are defined >>


  Commands:
   1) Create a new messaging server
   2) Edit the channel list for a messaging server
   3) Delete an existing messaging server
   q) Quit

Command to execute (1-3, q) : 1
  1) Publication broker
  2) Publication contractor
  3) Subscription contractor

Enter the number of the type you want to create ('q' to cancel):
```


To create a dedicated messaging server:

1. From the PeopleSoft Domain Administration menu, select the Messaging Server Administration menu.
2. From the Messaging Server Administration menu, select the *Create a new messaging server*.
3. From the submenu that appears, select the type of server to create.

   You can create a publication broker, a publication contractor, or a subscription contractor.
4. Enter a name to identify the new messaging server.

   The name is limited to six characters; for example, *PT8MSG.* The name that you enter is appended to each generic server process name; for example, *PSBRKDSP_PT8MSG* for the broker dispatcher and *PSBRKHND_PT8MSG* for the broker handler.

   Note: The name that you enter must be unique for the messaging server type in the current domain.
5. Specify the service operation queue that is handled by the new messaging server.

   You must specify a service operation queue, which must already be defined in the PeopleSoft Pure Internet Architecture.

   Note: The service operation queue name that you enter must exactly match the name that appears in the PeopleSoft Pure Internet Architecture. No prompt or validation occurs between PSADMIN and PeopleSoft Pure Internet Architecture definitions.

   Important! Don’t specify a given service operation queue for more than one messaging server of each type in the current domain. For example, you cannot have two subscription contractors assigned to the service operation queue. Nor can you have two dispatchers assigned to the service operation queue.

After several status messages, the Messaging Server Administration menu reappears, displaying a list of the existing dedicated messaging servers for the current domain.

Editing Messaging Server Queue Lists

After you create a publication broker, publication contractor, or subscription contractor, you may need to add more service operation queues to the server’s queue list, or you may want to decrease the number of service operation queues it services to improve performance.

This example illustrates the Messaging Server Administration menu. Use command *2* (Edit the queue list for a messaging server) to modify the queue list for a messaging server.

```
------------------------------------------------------------
             Messaging Server Administration menu
------------------------------------------------------------
  Domain Name : PT848805I1

  In addition to the default messaging servers, the following
  dedicated messaging servers are in the domain configuration:

   SERVER NAME       TYPE  QUEUES
   ----------------  ----  ----------------
   TEST01            BRK   QE_FLIGHTPLAN_QUEUE
   TEST02            PUB   QE_PO_QUEUE


  Commands:
   1) Create a new messaging server
   2) Edit the queue list for a messaging server
   3) Delete an existing messaging server
   q) Quit

Command to execute (1-3, q) : 2


       SERVER NAME       TYPE  QUEUES
       ----------------  ----  ----------------
   1)  TEST01            BRK   QE_FLIGHTPLAN_QUEUE
   2)  TEST02            PUB   QE_PO_QUEUE

Enter the number of the server to be edited: 2

Changing queue list for server 'TEST02'...

FORMAT: Alphanumeric, max 30-char queue names separated by commas (no tabs or spaces)
Current queues: [QE_PO_QUEUE]
Enter new queues:
```

To modify a queue list:

1. From the PeopleSoft Domain Administration menu, select **Messaging Server Administration menu.**
2. From the Messaging Server Administration menu, select Edit the queue list for a messaging server.
3. From the list of defined servers, select the messaging server for which you want to modify the queue list.
4. Specify a list of the message queues that will be handled by the selected server.

   You must specify at least one message queue. Multiple queue names must be entered as a list separated by commas, with no spaces; for example, *HRMS_01,HRMS_02,CRM_03*.

   Note: The new list of message queues that you enter replaces the current list of queues for the selected messaging server. The queues that you specify must already be defined in the PeopleSoft Pure Internet Architecture.

After several status messages, the Messaging Server Administration menu reappears, displaying the updated messaging server listing.

Deleting Messaging Servers

Sometimes a previously created messaging server is no longer needed. Rather than allow the server to consume valuable system resources, you should remove it from the domain.

To delete a messaging server from a domain:

1. From the PeopleSoft Domain Administration menu, select **Messaging Server Administration menu**.
2. From the Messaging Server Administration menu, select **Delete an existing messaging server**.
3. From the list of defined servers, select the messaging server to delete.

   After several status messages, the Messaging Server Administration menu reappears, displaying the remaining dedicated servers.

Configuring Messaging Servers

Once you create dedicated messaging servers, you must configure their dispatcher and handler processes so that they boot when you start the application server. You configure these processes using PSADMIN, as you do other server processes that run on the application server. Before you configure additional messaging server processes, familiarize yourself with the other server processes that run on the application server.

See “Understanding PSADMIN Menus” (System and Server Administration).

Two types of server processes comprise each messaging server: a dispatcher and a handler. Each process type requires that you set a different set of parameters. Most of the parameters are similar to other server processes, such as PSAPPSRV, but some parameters are specific to messaging servers.

Note: The following sections also apply to the _dflt messaging server processes. Only one parameter is different for a dedicated messaging server process and its _dflt counterpart—the Queues parameter. That parameter enables you to add message queues to the queue list. The _dflt server processes cannot be associated with a specific message queue.

Specifying Dispatcher Parameters

There are three generic process types that are the basis for all dispatcher processes:

- PSBRKDSP, which is the publication broker dispatcher.
- PSPUBDSP, which is the publication contractor dispatcher.
- PSSUBDSP, which is the subscription contractor dispatcher.

The following parameters apply to all three process types.

Field or Control	*Description*
Recycle Count	Specifies the number of times each dispatcher process is executed before being terminated (intentionally) by the system and then immediately restarted. **Note:** In general, you should not recycle dispatchers and should set this property equal to *0* (zero). The Recycle Count parameter does not translate into a native Oracle Tuxedo parameter in the PSAPPSRV.UBB file. Instead, the value is stored in memory and is managed by the system.
Allowed Consec Service Failures (Allowed consecutive service failures)	This option enables dynamic server process restarts in the event of service failures. To set this option, enter a number greater than *0*. To disable it, enter *0*. The default value for this parameter is *2*. The value that you enter is the number of consecutive service failures that cause a recycle of the server process. This is a catchall error handling routine that allows a dispatcher to terminate itself if it receives multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively; however, if they do, it indicates that the server process needs to be recycled or cleansed. A retry message appears when the specified number of service failures occurs.
Dispatch List Multiplier	Limits the number of dispatched messages by the number you specify, multiplied by the number of associated handler (s). This parameter is useful for unordered queues when all messages could go out at once. The default value is *10*.

Field or Control	*Description*
Scan Interval	Specifies the number of seconds between scans of the work queue when idle. The default value is *15* seconds. The scan interval is necessary to detect the following types of messages: • Messages published from an application server domain that is not the active pub/sub domain as selected on the Domain Status page in the Service Operations Monitor. • Cases where the broker server does not receive a notice of the publication. When a message is in the queue, the broker server doesn't receive a notice of the publication. A scan interval is required to make sure these types of messages are processed in a timely manner. The scan interval is analogous to the polling that PeopleSoft Process Scheduler performs on the Process Request table. In addition, the scan interval detects messages that have been resubmitted—for example, after an error. Decreasing the scan interval decreases latency for these types of publishes and error recovery. **Note:** The scan interval and ping rate (as a percentage) determines the actual interval for pinging any unavailable remote nodes. The algorithm used is: (attempts) x (ping rate) x (scan interval).
Ping Rate	Determines the number of seconds of inactivity before the server scans the database queues to restart any stalled or crashed items. The default value is *150* seconds. The ping rate is used in conjunction with the scan interval for pinging remote nodes. See the definition for Scan Interval in this section.
Maximum Ping Interval	Determines the maximum interval, in hours, between subsequent attempted pings of any unavailable remote nodes.
Dispatcher Queue Max Queue Size	Determines the maximum number of items per service operation queue that the dispatcher keeps in memory. The default value is *1000*.

Field or Control	*Description*
Memory Queue Refresh Rate	PeopleSoft Integration Broker maintains current asynchronous messaging queues in system memory for quick access. Occasionally, these cached queues can become corrupted. At that point, they must be refreshed from the PeopleSoft Integration Broker data tables. The likelihood and frequency of cache corruption depends on a combination of factors specific to the messaging system. If you need to periodically refresh the in-memory queues, you can use this parameter to tailor the frequency of the refresh to fit the situation. Each dispatcher on the system has its own queue. For each queue, you set the rate equal to the number of dispatch attempts that must occur before the queue is refreshed. The refresh occurs only when the specified number of dispatch attempts is reached for a given message queue. For example, with a memory queue refresh rate of *8*, multiple queues could have up to seven dispatch attempts each without triggering any refresh. The following settings are also significant: • A setting of *0* (the default) disables the refresh altogether. • A setting of *1* triggers a refresh immediately after every dispatch attempt, effectively disabling memory caching.

Field or Control	*Description*
Restart Period	Specifies the number of seconds between restart attempts on *Started* items in the work queue. An item which stays in *Started* state for more than a few seconds might be stalled—for example, the service request might have been lost, or the handler might have crashed. Decreasing the restart period reduces the latency for recovering stalled items with the status *Started*. However, under high load, items might stay in the *Started* state longer than normal for valid reasons. All handlers might be busy, and the handler service request for the item might be queued at the Oracle Tuxedo level. Setting the restart period too low results in redundant restarts. The dispatcher dispatches the item again, even though the original request is still in the Tuxedo queue. A small number of extra restarts is benign; however, at higher volumes, the unnecessary restarts can fill up the queue and block real requests. The formula for a reasonable value for the restart period is: ((incoming requests per second) / (number of handlers)) × (average processing time per request) For example, if you have an incoming rate of 20 per second, and you have four handlers, each handler is busy processing one item and will have four others waiting in the queue. A new item must wait for the currently processing item—plus the four items in the queue—before it is processed. If each item takes 10 seconds to process, the new item will stay in *Started* status for approximately 50 seconds before the handler works on it. If it stays in *Started* status longer, it's likely that the request to the handler has been lost, and the item should be restarted. **Note:** Using a value greater than *3540* for the dispatcher restart period results in constant restarts.

Specifying Messaging Server Process Handler Parameters

There are three generic process types that are the basis for all handler processes:

- PSBRKHND, which is the publication broker handler.
- PSPUBHND, which is the publication contractor handler.
- PSSUBHND, which is the subscription contractor handler.

The following parameters apply to all three process types.

Field or Control	*Description*
Min Instances (Minimum instances)	Specifies the number of handler server processes started at boot time.

Field or Control	*Description*
Max Instances (Maximum instances)	Specifies the maximum number of handler server processes that can be started or spawned.
Service Timeout	Specifies the number of seconds a handlers waits for a service request before timing out. Service timeouts are recorded in the TUXLOG and APPSRV.LOG. In the event of a timeout, the handler terminates itself and Oracle Tuxedo automatically restarts the process.
Recycle Count	Specifies the number of times that the system executes each server before the PeopleSoft system intentionally terminates the process. Server processes must be intermittently recycled to clear buffer areas. The time required to recycle a server is negligible (a matter of milliseconds). The Recycle Count parameter does not translate into a native Oracle Tuxedo parameter in the PSAPPSRV.UBB file. Instead the value is stored in memory and is managed by the PeopleSoft system.
Allowed Consec Service Failures (Allowed consecutive service failures)	This option enables dynamic server process restarts in the event of service failures. To set this option, enter a number greater than 0. To disable it, enter *0*. The default for this parameter is *2*. The numerical value that you enter is the number of consecutive service failures that cause a recycle of the server process. This is a catchall error handling routine that allows a handler to terminate itself if it receives multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively; however, if they do, it indicates that the server process needs to be recycled or cleansed. A retry message appears when the specified number of service failures occurs.
Max Retries (Maximum retries)	Specifies the maximum number of times that the server attempts to restart a failed action. This parameter prevents a bad item from continuously crashing a handler process. The counter is incremented when the handler sets the status to *Working* but before it actually starts processing the item.

Specifying the Disable Physical Document Cache Parameter

Use this parameter to disable cache for physical representations (json, xml).

Field or Control	*Description*
Disable Physical Document Cache	Disables the physical document cache. In the development environment, set this property to 1 to disable cache. In the production environment, set this property to 1 to disable cache. **Note:** Process requests will still be executed if this property is set to 1. However, setting this property to 0 in the production environment will enhance performance.

Related Links

"Integration Broker Options" (System and Server Administration)

Setting the Oracle Tuxedo Queue Size

The messaging system uses the Tuxedo queue size indicated in the application server domain section of PSADMIN to determine when the Tuxedo queue size has reached its maximum. The pub/sub system reads the actual queue size periodically, based on the Tuxedo Queue Status Check Count parameter. The system throttles itself so that it does not exceed this maximum, thereby preventing queue saturation and degraded performance.

Set the Tuxedo Queue Size parameter equal to that of the kernel parameter used by the machine running the pub/sub processes (msgsys:msgingo_msgmax).

To set the Tuxedo queue size for the messaging system:

1. In PSADMIN navigate to the **Values for config section – PSAPPSRV** part of the file. To do so:
    a. Open PSADMIN.
    b. Enter *1* for **Application Server** and press **Enter**.
    c. Enter *1* for **Administer a Domain** and press **Enter**.
    d. Choose a domain from the list and press **Enter**.
    e. Choose *4* for **Configure the Domain** and press **Enter**.
    f. Enter *Y* to shut down the domain.
    g. Enter *Y* to change the configuration values.
    h. Press **Enter** to scroll through the file and accept the current settings until you reach the following section:

```
Values for config section - PSAPPSRV
```


2. Enter *Y* and press **Enter** to change values in the section.
3. Navigate to the **Tuxedeo Queue Size** parameter. To do so, press **Enter** to scroll through the list and accept the current values. When you reach the Tuxedo Queue Size parameter enter a value.

   A value of *0* (zero) disables Tuxedo queue threshold determination and usage.

   Based on your environment, a value of *-1* sets the queue size to the following default values:

   - Windows: 65535.
   - AIX: 4000000.
   - Solaris: 65535.
   - HP: 65535.

4. Press **Enter** to scroll through the remaining sections and accept the current settings.

   PSADMIN will process the changes and then load the new configuration.
5. Boot the domain.

Related Links

“Understanding PSADMIN Menus” (System and Server Administration)

Chapter 4

Managing Integration Gateways

Understanding Managing Integration Gateways

This topic discusses managing integration gateways. This includes code examples to illustrate concepts and features of the integration gateway.

Note: The code examples in this topic are for illustrative purposes only and are not intended for use in a production environment.

Understanding Integration Gateway Configuration

This section discusses:

- Integration gateway versions and application server versions.
- Local gateway compatibility.
- Types of integration gateway configuration.
- The Gateways component.
- Minimum integration gateway setup requirements.

Integration Gateway Versions and Application Server Versions

Local and remote integration gateways must be at the same or higher version as the application servers with which they communicate.

Any remote gateway that you configure must be at the same version as the local-defined gateway.

Out of the box, PeopleTools 8.59 is delivered with a PeopleTools 8.59 integration gateway and a PeopleTools 8.59 application server, thus meeting this requirement.

However, situations may arise where your integration environment is comprised of PeopleSoft systems running different versions of PeopleTools, or your integration partners are running different versions of PeopleTools. When this is the case, you must ensure that the integration gateway is at the same or higher version as the application server with which it communicates, or integrations will fail.

The following list describes several compatible integration gateway/application server version combinations:

- You are using a PeopleTools 8.59 integration gateway to communicate to a PeopleTools 8.59 application server.

- You are using a PeopleTools 8.59 integration gateway to communicate to a PeopleTools 8.58 or earlier application server.
- You are using a PeopleTools 8.59 remote integration gateway to communicate with a PeopleTools 8.58 or earlier application server.

The following list describes several incompatible integration gateway/application server version combinations. Integrations will fail with these combinations:

- You are using a PeopleTools 8.59 or earlier integration gateway to communicate with a PeopleTools 8.60 application server.
- You are using a PeopleTools 8.59 remote integration gateway to communicate with a PeopleTools 8.60 application server.

Local Gateway Compatibility

Because database administrator passwords and gateway keystore passwords are encrypted in the current PeopleTools release, the local gateway specified by a node in the current release of PeopleSoft Integration Broker must be from PeopleTools 8.55 or later version to support encryption. If you upgrade PeopleTools and the integration engine is from a release earlier than PeopleTools 8.55, you must also upgrade the local gateway.

Note: The current release of the integration gateway works with nodes that use PeopleTools 8.4x or later PeopleSoft Integration Broker.

Types of Integration Gateway Configuration

An integration gateway requires several types of configuration:

Term	***Definition***
Security configuration	You can implement PeopleSoft Integration Broker security in several ways. At a minimum you must install digital certificates on the machine on which the gateway is installed. Complete the certificate installation before continuing with the gateway configuration in this topic. Once the gateway's digital certificates are installed, you must enter several configuration parameters in the Integration Gateway Certificates Section of the integrationGateway. properties file. The parameters you must set are the certificate alias name, the certificate alias password, the path to the keystore, and the keystore password. See Installing Digital Certificates on the Integration GatewayInstalling Integration Gateway-Based Digital Certificates.

Term	*Definition*
General configuration	This includes settings for the gateway version, class location, general communication parameters, node connection parameters, message and error logging, and gateway type and location. Most of these settings are entries in the integrationGateway.properties file, but you set a few of them in the Gateways component.
Connector-specific configuration	The number of configuration settings and where they're applied depend on the connector. You configure most of the target connectors delivered with PeopleSoft Integration Broker by using the Gateways component, but some require settings in the integrationGateway.properties file. A few require settings in both environments. **Note:** You can override some target connector properties for an individual node.

The Gateways Component

Once the gateway has been installed, you use the Gateways component (IB_GATEWAY) to make it accessible to any node that uses it for messaging. You can also use it to override the gateway's default connector properties for individual nodes without having to directly edit the integrationGateway.properties file on the gateway machine.

See Understanding Managing Integration Gateways.

Minimum Integration Gateway Setup Requirements

The minimum setup requirement to run an integration gateway are:

1. Specify the gateway URL.

   See Defining Integration Gateways.

2. Specify the Oracle Jolt connection string properties to enable communication with each PeopleSoft Integration Broker node that will be involved in an integration that uses a gateway.

   See Setting Oracle Jolt Connection Properties.

3. Set and encrypt the keystore password.

   See Configuring Security and General Properties.

Administering Integration Gateways

This section discusses how to:

- Define integration gateways.

- Ping integration gateways.
- Load target connectors.
- Edit connector properties.

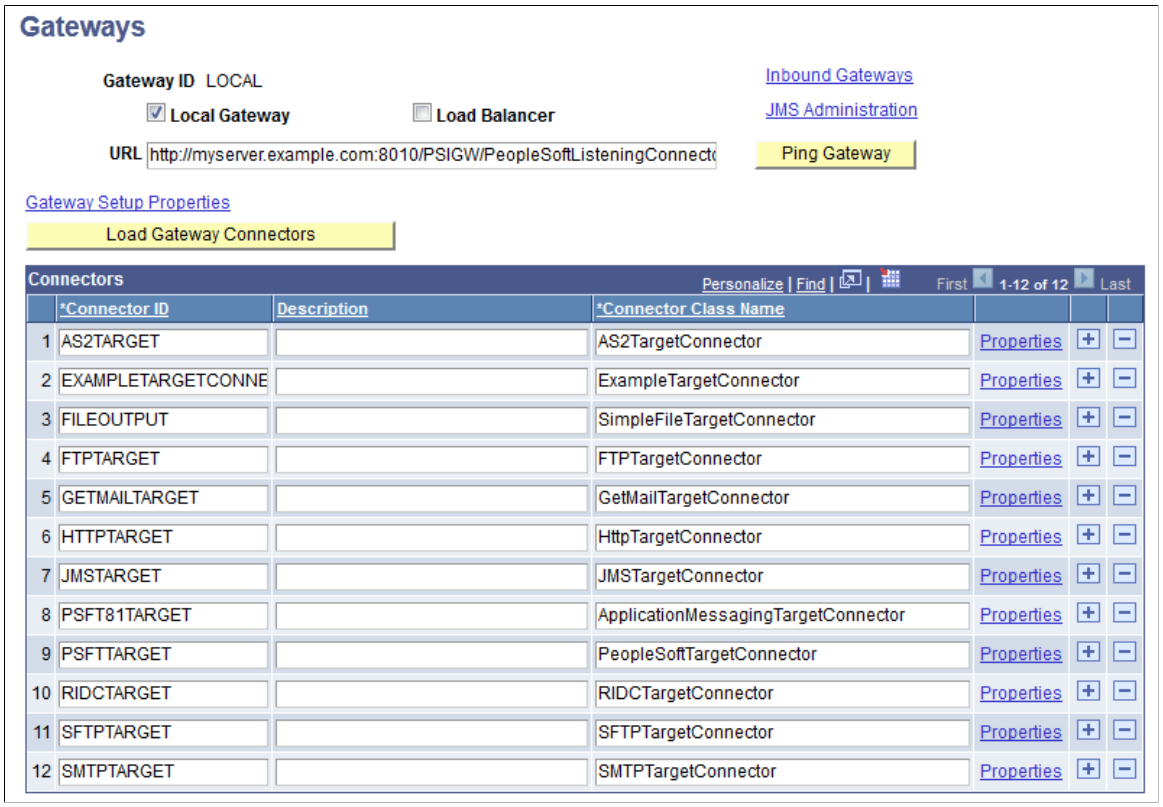
Defining Integration Gateways

Use the Gateways page (IB_GATEWAY) in the Gateways component (IB_GATEWAY) to specify the location of the gateway, update configuration settings, and register target connectors to be used with the gateway.

Note: A default local gateway definition is automatically created upon installation. If you plan to use only the local gateway, you do not need to create a new definition; however, you still must configure the gateway.

To access the Gateways page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**.

This example illustrates the Gateways page with the gateway URL defined and the target connectors loaded.



To define and configure a gateway:

1. Access the Gateways page (select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**).
    - Click **Search**, and select an existing gateway definition.

The Gateways page appears, displaying the gateway definition.

Note: The default ID for the delivered local gateway is *LOCAL*.

- Add a new value, enter an integration gateway ID, and click **Add**.

  The Gateways page appears.

2. (Optional.) Select *Local Gateway* to designate the gateway as local.

   Each PeopleSoft Integration Broker node requires exactly one local gateway, which is the application's first point of contact with other PeopleSoft applications, third-party systems, Integration Broker hubs, and remote gateways.

   Note: You must open the definition of the designated local gateway and clear the Local Gateway check box before you can select that check box in another definition.

3. Enter the gateway URL for the selected gateway's PeopleSoft listening connector.

   Specify the URL with the format:

   http://*machinename:port*/PSIGW/PeopleSoftListeningConnector

   In this case, *machinename:port* is the machine name and port, host name, or IP address of the web server hosting the gateway.

   By default the port number is *80* for HTTP and *443* for HTTPS. If using the default port number, you do not need to specify it in the URL.

   For HTTPS, the URL should start with *https*.

   The integration gateway URL is case sensitive.

   The gateway uses the PeopleSoft listening connector to receive service operations from an integration engine node or a remote gateway.

4. (Optional.) To load the delivered target connectors, click the **Load Gateway Connectors** button.

   You can load the delivered target connectors at this point, or at a later time.

5. Save the gateway definition.

6. Click the **Gateway Setup Properties** link to configure additional gateway settings and connector properties.

Related Links

Configuring Integration Gateways for Load Balancing When Using Third-Party Software

Pinging Integration Gateways

Use the Gateways page to ping an integration gateway to verify that it is running. Before you ping an integration gateway, you must define the gateway URL.

To ping an integration gateway:

1. Access the Gateways page (select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**).
2. Select the integration gateway to ping.

   The Gateways page appears.
3. Click the **Ping Gateway** button.

If the ping is successful a PeopleSoft Listening Gateway page appears that displays a status of *Active*.

Loading Target Connectors

The Connectors grid on the Gateways page lists the target connectors registered with the current gateway. Initially, none of the delivered connectors are loaded and the grid is empty. You can load target connectors automatically by introspection or manually by entering information in the grid.

Note: You typically load and configure the gateway target connectors only when you configure a new gateway or install a new connector.

Loading Connectors by Introspection

If the connector was delivered with the PeopleSoft application or developed using the PeopleSoft Integration Broker Connector Software Development Kit (SDK), you can easily load it with the PeopleSoft Integration Broker connector introspection feature. Before you can register a new connector, you must install it.

See Understanding the PeopleSoft Integration Broker Connector SDK.

To load connectors by introspection:

1. Access the Gateways page (select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**).
2. Click the **Load Gateway Connectors** button to trigger introspection for the current gateway.

PeopleSoft Integration Broker examines the properties of all installed target connectors and loads those properties into the gateway definition. All the connectors appear in the Connectors grid, and the properties of each connector are updated to reflect its current state.

Note: The introspection never overrides existing information. It adds only missing information, so manually edited values are not affected. If you modified a connector, new and modified properties are loaded and do not interfere with existing properties.

Loading Connectors Manually

To load and configure a connector manually, you enter the connector ID, connector class name, and property information that's hard-coded in the connector. This information is provided by PeopleSoft for all delivered connectors; information about connectors from any other source must be provided by that source.

To load a new connector manually:

1. Access the Gateways page (select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**).
2. Add a new row in the Connectors grid.
3. Enter the ID for the new connector.
4. Enter the connector class name.
5. Click **Properties** to edit the connector's properties.

Related Links

Understanding Using the Connector Examples
Understanding the PeopleSoft Integration Broker Connector SDK

Editing Connector Properties

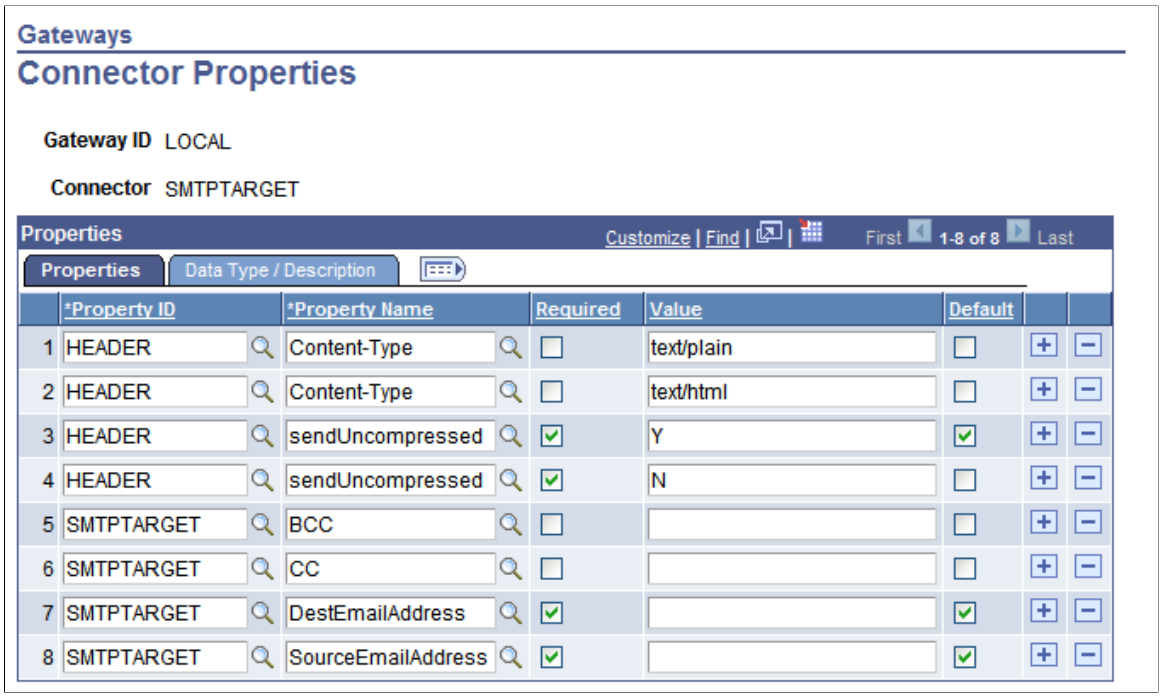
Node-level target connector properties represent parameters that can be used by the connector. These properties are hard-coded in the connector class. The Connector Properties page (IB_CONNPROP) lists all of a connector's available properties and their values. When you specify a connector in a node definition, only the properties that you are required to set and specify display.

Note: Available connector properties are automatically entered on the Connector Properties page when you register the connector.

Each property entry is defined by a combination of property ID and property name, both of which must already exist in the connector class. A single connector can handle service operations that adhere to different header formats, communication protocols, or other requirements. You can represent these variations on the Connector Properties page by entering multiple instances of the properties used, each with a different value.

Warning! Do not add new properties to any of the delivered connectors, as doing so requires changes to the delivered Java connector programs. Add connector properties only for custom connectors you have created.

This example illustrates the Connector Properties page and the properties for the SMTPTARGET connector.



To add a new property instance:

1. Access the Connector Properties page (select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** to display the Gateways page).
2. In the Connectors section locate the row that lists the target connector with which you want to work, and click the Properties link at the end of the row. The Connector Properties page displays.
3. Select a **Property ID**.

   Available property IDs are specific to the connector that you're configuring.
4. Select a **Property Name**.

   The available property names are specific to the property ID that you selected.
5. If the property is required for the connector to work properly, select the **Required** check box.

   All instances of a property (that is, all identical property ID and property name combinations) should have the same Required status.
6. Enter an appropriate value for the property instance.

   Appropriate values might come from PeopleSoft, from the connector's developer, or from your own experience and requirements.
7. (Optional.) Select the **Default** check box.

   When you specify the connector in a node definition, only properties marked as both required and default appear automatically on the Connectors page of the Node Definitions component.

Note: In most cases, only one instance (value) of a required property should be used by a given node; however, you might designate multiple values as default so that they all appear. Keep in mind which properties can be used with multiple values and which ones require a single value.

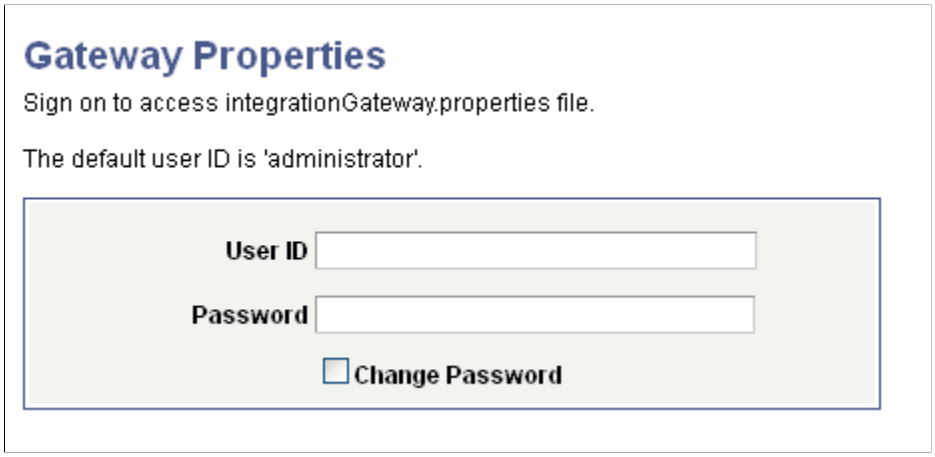
8. Save the properties.
9. Click **OK**.

   The Gateways page appears.

Accessing Gateway Setup Properties

To access gateway setup properties from the Gateways page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Gateway Setup Properties** link. The Gateway Properties sign in page (IBGWSIGNON) appears.

This example illustrates the Gateway Properties page.



The default user ID is *administrator* and the default password is set during the installation of the PeopleSoft Pure Internet Architecture.

After you successfully enter the user ID and password, the PeopleSoft Node Configuration page displays where you specify information about how to connect to nodes and access the integrationGateway.properties file to establish additional gateway settings.

Changing the Gateway Properties Access Password

To change the gateway properties access password:

1. In the **User ID** field, enter the user ID.
2. In the **Password** field, enter the existing password.
3. Select the **Change Password** box.

   A **New Password** field and a **Confirm Password** field appear.

4. In the **New Password** field, enter a new password.
5. In the **Confirm Password** field, enter the new password again.
6. Click the **Save** button.

Resetting the Gateway Properties Access Password

You can reset the password in the gatewayUserProfile.xml file located in <PIA_HOME>\webserv \<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF. The password you enter in the gatewayUserProfile.xml file must be encrypted. Use the PSCipher utility to encrypt the password.

Setting Oracle Jolt Connection Properties

The integration gateway communicates with PeopleSoft application server nodes using Oracle Jolt connections.

Understanding Oracle Jolt Connection Properties

This section discusses setting Oracle Jolt connection string properties using the PeopleSoft Node Configuration page. Setting these properties in the integrationGateway.properties file is discussed later in this section.

The PeopleSoft Node Configuration page (PSGTWPROPS_SEC) provides grids for defining Oracle Jolt connection properties for unknown (default) and known nodes. When you save the properties you set on this page, they are written to the integrationGateway.properties file. To edit or define these properties in the future, you can use the PeopleSoft Node Configuration page or the integrationGateway.properties file.

Connection Settings When Target Nodes are not Known

Within any inbound message, the integration gateway requires only the names of the message and the requesting node. If the message is sent by a PeopleSoft Integration Broker system, it also includes the name of the target node. The gateway searches the integrationGateway.properties file for the Jolt connect string properties for the specified target node, so it can properly direct the message.

However, the integration gateway cannot determine the target node in the following cases:

- The Jolt connect string settings for the specified target node are missing from the integrationGateway.properties file.
- The message format does not include a To node specification.

To handle these cases, you can specify a default application server to handle the message if no valid target node can be determined.

Connection Settings for Known Target Nodes

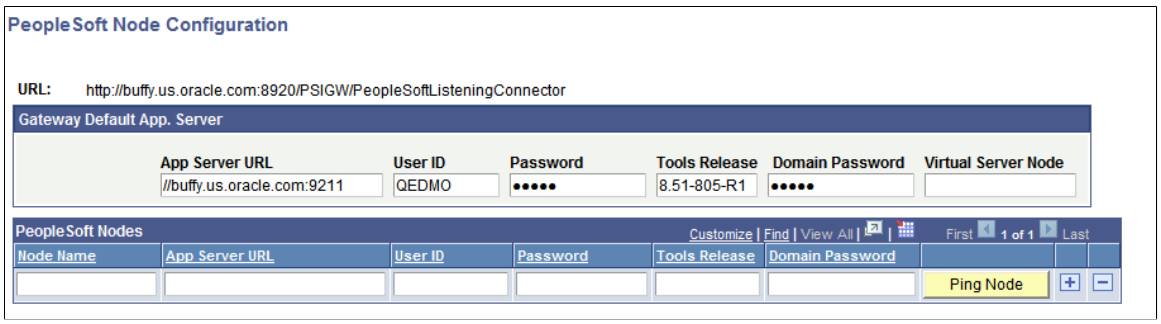
You must set four Oracle Jolt connect string properties for each PeopleSoft Integration Broker application server node with which the integration gateway communicates. The gateway uses this information to access each node's database through a Oracle Jolt connection with its PeopleSoft target connector.

Note: These properties apply only to communications that don't cross a firewall and for which the gateway uses the PeopleSoft target connector.

Setting Oracle Jolt Connection String Properties

The PeopleSoft Node Configuration page provides a grid for setting Oracle Jolt connection string properties for unknown (default) target nodes and known target nodes.

This example illustrates the fields and controls on the PeopleSoft Node Configuration page. You can find definitions for the fields and controls later on this page.



To access the PeopleSoft Nodes Configuration page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways.** The Gateways page appears. Click the **Gateway Setup Properties** link. Enter the gateway user ID and password and click the **OK** button. The PeopleSoft Node Configuration page appears.

To define properties for unknown nodes use the Gateway Default Application Server grid on the PeopleSoft Node Configuration page. To define properties for known nodes use the PeopleSoft Node grid on the PeopleSoft Node Configuration page.

Note: Setting Oracle Jolt string connection properties for unknown nodes is optional.

Field or Control	***Description***
App Server URL(Application Server URL in the Gateway Default App Server Section)	Enter the machine name and Oracle Jolt port number of the default application server to use if no valid target node can be determined. To determine the Jolt port of the application server, check the JOLTListener section in the psappsrv.cfg file. The file is located in *<PS_CFG_HOME>\appserv\<DOMAIN_NAME>*.
App Server URL(Application Server URL in the PeopleSoft Nodes Section)	Enter the machine name and Oracle Jolt port number of the default application server to use if no valid target node can be determined. **Note:** To determine the Jolt port of the application server, check the JOLTListener section in the psappsrv.cfg file. The file is located in *<PS_CFG_HOME>\appserv\<DOMAIN_NAME>*.

Field or Control	*Description*
Domain Password	Enter the password for the domain as entered in PSADMIN.
Node Name	Enter name of the PeopleSoft node with which the integration gateway is to communicate.
User ID	Enter the user ID that you defined when you created the application server domain.
Password	Enter the user password that you defined when you created the application server domain. PeopleSoft Integration Broker will automatically encrypt this password entry.
Tools Release	Enter the PeopleTools version number installed on the application server. If you are installing a patch build, include the patch number. For example, if you are installing PeopleTools 8.55 patch build 3, enter the following: *8.55.03*
Virtual Server Node	This field is used in conjunction with inbound request processing using virtual server domains. See Implementing Inbound Request Load Balancing Using Virtual Application Server Domains. Enter a node name from the list of nodes in the PeopleSoft Nodes grid on the bottom of the page. When utilizing virtual server domains, the system routes inbound requests that do not specify a “To” node to the this node.

The properties and values you set in the PeopleSoft Node Configuration page are located in the DELIVERED CONNECTOR CONFIGURATION Section of the integrationGateway.properties file.

The properties you set for unknown nodes are in the subsection ## JOLT connect string setting for optional Default Application Server. The properties you set for known nodes are in the subsection ## JOLT connect string settings for Application Server(s) with known NODENAMEs.

Related Links

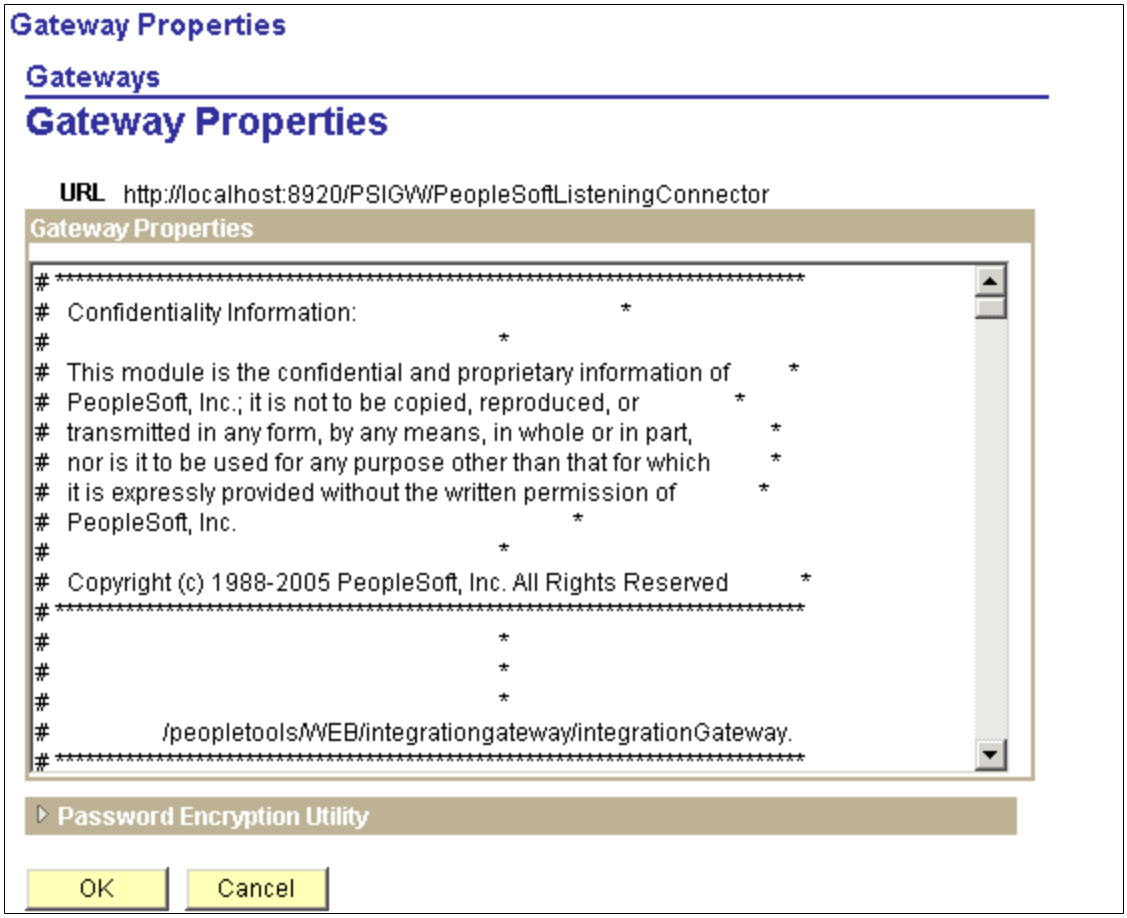
Accessing Gateway Setup Properties
Using the integrationGateway.properties File
Configuring Security and General Properties

Using the integrationGateway.properties File

To establish settings for the integration gateway and its delivered connectors, you use the integrationGateway.properties file.

The integrationGateway.properties file is a text file.

This example illustrates the Gateway Properties page.



The property settings in the file are stored as name-value pairs in labeled sections, and the lines are commented out using the pound sign (#). Here's an example of a commented-out property setting:

```
#ig.isc.userid=MYUSERID
```

Accessing the integrationGateway.properties File

You can access and edit the integrationGateway.properties file using the Gateways component in the Pure Internet Architecture or using the text file located in the *<PIA_HOME>\webserv* directory.

Understanding Accessing the integrationGateway.properties File

Most integration systems are configured such that the application server, integration gateway, and PeopleSoft Pure Internet Architecture are running the same PeopleTools versions.

However, there may be some instances where the integration system is configured such that there is a shared integration gateway working with application servers on different versions of PeopleTools. If this is the case, you must access and edit the integrationGateway.properties in one of the following ways:

- Manually edit the text file located in the <*PIA_HOME*> directory. Use the PSCipher Java utility to generate encrypted passwords.

  Using the PSCipher Java utility is described elsewhere in this topic.

  See Encrypting Passwords Using the PSCipher Java Utility.

- Access and edit the properties file via the PeopleSoft Pure Internet Architecture in the Gateways component. You must log into the PeopleSoft Pure Internet Architecture that is installed on the application server that is running the same PeopleTools release as the integration gateway. If you use this method, you must ensure that copies of the same psvault key file are installed on all application servers and gateway/web server in the configuration.

  The psvault key file is discussed elsewhere in the product documentation.

  See “Securing the External Key File” (Security Administration), “Generating a Unique Encryption Key” (Security Administration), “Updating the Encryption Key on Oracle WebLogic” (Security Administration),

Accessing the integrationGateway.properties File in the Pure Internet Architecture

Access to the integrationGateway.properties file using the PeopleSoft Pure Internet Architecture is password-protected.

To access the integrationGateway.properties file:

1. Select **PeopleTools** > **Integration Broker** > **Configuration** > **Gateway**.
2. Select a gateway with which to work.
3. Click the **Gateway Setup Properties** link.

   The Sign on to access the integrationGateway.properties file box displays.
4. Enter the user ID and password and click the **OK** button.
5. Click the **Advanced Properties Page** link.

The Gateway Properties page also provides access to the Password Encryption Utility and you can encrypt passwords required in the integrationGateway.properties file directly from that page.

Accessing the integrationGateway.properties File in the <PIA_HOME> Directory

The integrationGateway.properties file is located in the following path in the PeopleSoft home directory:

<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF \integrationGateway.properties.

When you access the integrationGateway.properties file in directly in the <*PIA_HOME*> directory, you must restart the PeopleSoft Pure Internet Architecture for the changes to take effect.

Related Links

Loading Target Connectors
Encrypting Passwords

Entering Values in the integrationGateway.properties File

When entering values in the integrationGateway.properties file that contain paths, you must use either double backslashes (“\\”) or forward slashes (“/”) as path separators.

Note: Do not use backslashes (“\”) as path separators for directory names in the integrationGateway.properties file. Backslashes are misinterpreted as escape characters by the Java processes that access the file.

To correctly specify a path in the integrationGateway.properties file, you must use either double backslashes (“\\”) or single forward slashes (“/”) as separators; for example:

```
ig.transform1.XSL=C:\\XSLProgs\\MyTransform.xsl
ig.transform1.XSL=C:/XSLProgs/MyTransform.xsl
ig.transform1.XSL=/usr/xsls/MyTransform.xsl
```

Note: The one exception to this is when entering path separators for EIP test automation properties. When working with those properties you must enter path separators as backslashes.

Encrypting Passwords

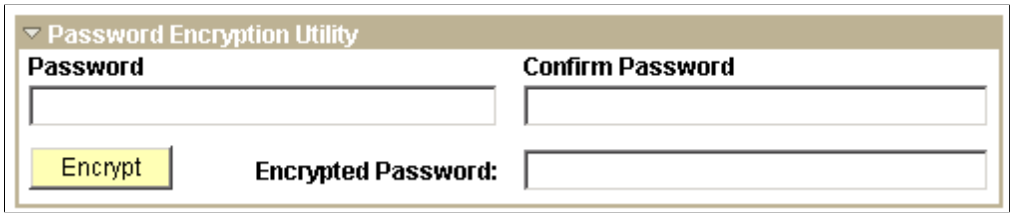
The integration gateway properties file and target connectors feature required and optional passwords. All passwords must be encrypted.

PeopleSoft provides an encryption utility, PSCipher, that you can use to encrypt passwords. You can access the utility from the PeopleSoft Pure Internet Architecture or from a Java utility.

Encrypting Passwords in the PeopleSoft Pure Internet Architecture

The Password Encryption Utility dialog box displays in areas where required or optional passwords are specified.

This example illustrates the Password Encryption Utility dialog box. Use the utility to encrypt passwords.



To encrypt a password using the Password Encryption Utility:

1. On the page where you are working, click the **Password Encryption Utility** arrow to display the dialog box.

2. In the **Password** field, enter a password.
3. In the **Confirm Password** field, enter the password again.
4. Click the **Encrypt** button. The encrypted password displays in the **Encrypted Password** field.
5. From the **Encrypted Password** field, cut the encrypted password and paste it into the appropriate location.

Encrypting Passwords Using the PSCipher Java Utility

You launch the PSCipher utility from the <PIA_HOME> directory.

To encrypt a password:

1. Launch the PSCipher.bat file in the <PIA_HOME>\webserv\<DOMAIN>\piabin directory.
2. If using UNIX, change the script file's permissions so that you can execute it.
3. Execute the script file with your password as an argument.

   The utility returns the encrypted password as a string.

   - On a Windows machine, enter:

     ```
     pscipher MYPASSWORD
     ```

   - On a UNIX machine, enter:

     ```
     PSCipher.sh MYPASSWORD
     ```

4. Copy the encrypted string and paste it into the appropriate location.

Configuring Security and General Properties

This section discusses how to:

- Set gateway digital certificate security properties.
- Specify the gateway version.
- Specify the gateway class location.
- Set general connection properties.
- Set logging properties.
- Set DTD validation properties.
- Set Oracle Jolt session pooling parameters.
- Set the namespace for generic SOAP faults.

- Display the PeopleTools version of an integration gateway.

Setting Gateway Digital Certificate Security Properties

You can implement gateway-based digital encryption on the integration gateway. When implemented you must set the integration gateway digital certificate alias, certificate alias password, keystore path, and keystore password in the integration gateway properties file.

At a minimum you must specify the path and password to the integration gateway keystore in the integration gateway properties file. Although highly recommended, you do not need to install digital certificates to meet this requirement. You need only specify the path and password to the keystore.

Use the PSKeyManager utility delivered with PeopleTools to generate the keystore password. See "Understanding PSKeyManager Utility" in "Installing Web Server-Based Digital Certificates" (Security Administration) for information on using the PSKeyManager utility.

Most production environments have SSL/TLS implemented. If the integration gateway is installed on a web server that has SSL/TLS implemented, you can specify the SSL/TLS keystore path and password to meet this requirement.

Important! SSL/TLS encryption applies digital certificates from two keystores to encrypt inbound and outbound messages, respectively. The integration gateway manages the certificates in the keystore that supports outbound messaging. So if you specify the SSL/TLS keystore path and password, be sure to use the proper values.

Use the section labeled *Integration Gateway CERTIFICATE Section* in the integration gateway properties file to define the keystore path and encrypt the keystore password.

Warning! Integrations will fail if you do not set the path to the keystore using the secureFileKeystorePath property and enter an encrypted keystore password for the secureFileKeystorePasswd property.

You must set the following properties in integrationGateway.properties so that the gateway can access the encryption certificates.

Property	***Description***
ig.certificateAlias	(Optional.) If implementing gateway-based digital certificates, enter the name that you provided to identify the encryption key pair that you generated for the keystore on which the gateway's public key certificate is based.
ig.certificatePasswd	(Optional.) If implementing gateway-based digital certificates, enter the password that you provided for the encryption key pair that you generated for the keystore. The certificate password must be encrypted. See Encrypting Passwords.
secureFileKeystorePath	Enter the full path and file name of the gateway keystore file, which is located in the web server directory structure. The path is <PIA_HOME>\webserv\<DOMAIN>\piaconfig\keystore.

Property	*Description*
secureFileKeystorePasswd	Enter the keystore password. . This password must be encrypted. See Encrypting Passwords.

Related Links

Installing Digital Certificates on the Integration Gateway
Installing Integration Gateway-Based Digital Certificates

Specifying the Gateway Version

The gateway version property, ig.version, indicates the version of PeopleTools from which the integration gateway is installed.

The integration gateway version must be the same or higher version than the version of the application server with which you want to communicate. For example, you can use a PeopleTools 8.55 integration gateway to communicate with a PeopleTools 8.55 application server or to communicate with a PeopleTools 8.47 application server.

Integration gateways cannot communicate with application servers on higher versions. For example a PeopleTools 8.47 integration gateway cannot directly communicate with a PeopleTools 8.55 application server. If this kind of communication is required, then the communication should be setup where the 8.47 gateway is set up as a remote gateway.

The version property is located in the integrationGateway.properties file in the section labeled Integration Gateway VERSION Section. Specify the version as follows:

```
ig.version=version_number
```

where *version_number* is the version of PeopleTools with two decimal places; for example, *8.55*.

Setting General Connection Properties

This section discusses:

- Default connector properties.
- Node-specific Oracle Jolt connect string properties.
- Default Oracle Jolt connect string properties.

The general connection properties include default connector properties and Oracle Jolt connect strings for nodes that designate this gateway as their local gateway. You can find these properties in the section of the integrationGateway.properties file labeled *DELIVERED CONNECTOR CONFIGURATION Section.*

Default Connector Properties

Property	*Description*
ig.connector.prefix	Identifies the universal resource indicator (URI) prefix added to any target connector name. This property instantiates the connector classes on the system. The default connector prefix is: `com.peoplesoft.pt.integrationgateway.targetconnector.` **Note:** Do not change this value.
ig.connector.defaultremoteconnector	Identifies the connector that the gateway uses to send messages to a remote gateway. The default value of this property is: `HttpTargetConnector` **Note:** Do not change this value.
ig.connector.ibtargetconnector	Identifies the connector that the gateway uses by default to send messages to a PeopleSoft Integration Broker application server node. The gateway uses this connector to link to the integration engine running on the node's application server. When the content of a message reaching the gateway doesn't specify a connector (this is often the case with third-party senders), the gateway automatically uses the connector specified by this property. The default value is: `PeopleSoftTargetConnector` **Note:** Do not change this value.

Default Oracle Jolt Connect String Properties

Within any inbound message, the integration gateway requires only the names of the message and the requesting node. If the message was sent by a PeopleSoft Integration Broker system, it also includes the name of the target node. The gateway searches the integrationGateway.properties file for the Jolt connect string properties for the specified target node, so it can properly direct the message.

However, the integration gateway cannot determine the target node in the following cases:

- The Jolt connect string settings for the specified target node are missing from the integrationGateway.properties file.
- The message format does not include a To node specification.

  This can include general HTTP calls to listening connectors other than the PeopleSoft listening connector.

- When using Send Master for testing purposes.

To handle these cases, you can specify a default target node for the gateway if no valid target node can be determined.

Note: You can set these properties in the integration gateway properties file or use the PeopleSoft Node Configuration page in PIA.

Use the default Jolt connect string properties:

```
#ig.isc.serverURL=//<machine name>:<jolt port>
#ig.isc.userid=<database user id>
#ig.isc.password=<database password>
#ig.isc.toolsRel=<peopletools release version>
```

Uncomment these four lines and enter values to designate a PeopleSoft Integration Broker node as the gateway's default (backup) target node. It typically is one of the nodes for which you already created node-specific Jolt connect string properties.

There's only one set of these default properties. They specify the same parameters as the node-specific properties, except that you don't include a node name; for example:

```
ig.isc.serverURL=//MYSERVER01:9000
ig.isc.userid=<database user id>
ig.isc.password=<database password>
ig.isc.toolsRel=8.55
```

You can also specify a comma-separated list of server URLs. Each URL in the string will use the user ID, password, and PeopleTools release defined:

```
ig.isc.serverURL=//MYSERVER01:9000,//MYSERVER02:9250,//MYSERVER03:9500,//MYSERVER04⇒

:9750
ig.isc.userid=<database user id>
ig.isc.password=<database password>
ig.isc.toolsRel=8.55
```

Oracle Jolt Connect String Properties for Known Nodes

You must set four Oracle Jolt connect string properties for each PeopleSoft Integration Broker application server node with which the integration gateway communicates. The gateway uses this information to access each node's database through a Oracle Jolt connection with its PeopleSoft target connector.

Note: You can set these properties in the integration gateway properties file or use the PeopleSoft Node Configuration page in PIA.

Note: These properties apply only to communications that don't cross a firewall and for which the gateway uses the PeopleSoft target connector.

The integrationGateway.properties file contains a template for these properties:

```
ig.isc.$NODENAME.serverURL=//<machine name>:<jolt port>
ig.isc.$NODENAME.userid=<application server user id>
ig.isc.$NODENAME.password=<application server password>
ig.isc.$NODENAME.toolsRel=<peopletools release version>
```

For each node, make a copy of this template and replace *$NODENAME* with the name of the node definition. Enter appropriate values for each property as described in the following table:

Property	*Description*
ig.isc.$NODENAME.serverURL	Enter the URL of the application server node, consisting of the machine name and Oracle Jolt port; for example: `ig.isc.MYNODE.serverURL=//MYMACHINE:9000` **Note:** You can determine the Jolt port of the application server by examining the *JOLT Listener* section in the psappsrv.cfg file located in <PS_CFG_HOME>\appserv\<DOMAIN_NAME>.
ig.isc.$NODENAME.userid	Enter the User ID that you defined when you created the application server domain.
ig.isc.$NODENAME.password	Enter user password that you defined when you created the application server domain. This password must be encrypted. See Encrypting Passwords.
ig.isc.$NODENAME.toolsRel	Enter the version number of PeopleTools installed on the application server node to two decimal places; for example: `ig.isc.MYNODE.toolsrel=8.55`

You can also specify a comma-separated list of server URLs. Each URL in the string will use the user ID, password, and PeopleTools release defined:

```
ig.isc.QEDMO.serverURL=//MYSERVER20:9000,//MYSERVER21:9100,//MYSERVER23:9200,//MYSE⇒

RVER24:9300
ig.isc.QEDMO.userid=<user ID>
ig.isc.QEDMO.password=<password>
ig.isc.QEDMO.toolsRel=8.55
```

Setting Logging Properties

This section discusses:

- General logging properties.
- Message logging properties.
- Error logging properties.
- Overriding the IP address used for gateway logging.

The logging properties specify parameters for logging messaging activity and errors. You can find these properties in the section of the integrationGateway.properties file labeled *LOGGING Section.*

General Logging Properties

Property	*Description*
ig.log.level	Enter a numeric value to specify the desired level of gateway logging and exception handling. Values are: • *−100:* Suppresses message logging. The property is preset to this value. • *−1:* Logs language exceptions only. • *1:* Logs language and standard exceptions. • *2:* Logs all errors and warnings. • *3:* Logs errors, warnings, and important information. This is the default if you don't specify a value for this property. • *4:* Log errors, warnings, and important and standard information. • *5:* Logs errors, warnings, and important, standard, and low-importance information. **Note:** Set the log level to *5* to capture the entire contents of incoming HTTP requests, including HTTP headers, in the integration gateway message log file.
ig.log.backgroundImage	Specify the background image to use when displaying error and message log documents. The image must be in jpg format. The default location and image name PSbackground.jpg. By default it is located in <PIA_HOME>\webserv \<DOMAIN>\applications\peoplesoft\PSIGW.war. Images in the default location don't require a path, but you can specify a full path to an image file in any other location.

Message Logging Properties

Property	*Description*
ig.messageLog.filename	Enter the full path and file name of an HTML file to use as a message log. This property is preset to <PIA_HOME> \webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war \WEB_INF\msgLog.html.

Property	*Description*
ig.messageLog.maxSize	Specify the maximum size of the message log, in kilobytes (KB). This property is preset to *10000,* or 10 megabytes (MB). When this limit is reached, the log is archived, and a timestamp is appended to the file name.
ig.messageLog.maxNbBackupFiles	Specify the number of archived files to keep on disk. Use the value *0* to retain all backed up files. This property is preset to *5*.

Error Logging Properties

Property	*Description*
ig.errorLog.filename	Enter the full path and file name of an HTML file to use as an error log. This property is preset to <PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF\errorLog.html.
ig.errorLog.maxSize	Specify the maximum size of the error log in kilobytes (KB). This property is preset to *1000,* or 1 MB. When this limit is reached, the log is archived, and a timestamp is appended to the file name.
ig.errorLog.maxNbBackupFiles	Specify the number of archived error files to keep on disk. Use the value *0* to retain all backed up files. This property is preset to *5*.

Related Links

“Understanding Error Handling, Logging, Tracing and Debugging” (Integration Broker)

Overriding the IP Address Used for Gateway Logging

The gateway captures its own IP address and returns this information in responses sent to the application server. The application server uses the address to locate the physical machine where the gateway runs; this is where the various gateway logs are stored.

In certain instances, the value of the IP address captured may be incorrect, due to issues with the underlying web server. For example, the address may show up as 192.0.2.10, which from the gateway's perspective is correct (since that's the IP loopback address) but this value is not useful to the application server as it needs the actual IP of the gateway.

PeopleSoft provides the following property that enables you to override the IP address that the gateway captures and hardcode the IP address used for gateway logging:

```
ig.GatewayIPAddressOverride
```

This property is located in the Gateway IP Address Override section in the integrationGateway.properties file.

To use the property, set it equal to the IP address of the machine running the integration gateway.

Note that the actual machine that writes the logs does not change. If set the value to one that is not the IP address of the machine that the gateway is actually on, when the application server creates links to display log info, those links will not point to the actual log files.

Setting DTD Validation Properties

You can validate XML request messages and response messages against associated document type definitions (DTD) by enabling DTD validation on the integration gateway.

When you set the ig.dtdLookup property equal to *True* (default), request and response messages are validated against any associated DTD.

References to DTDs may be inline pointers to files or references to URLs.

When you set the ig.dtdLookup property equal to *False*, no validation takes place—even if a DTD reference is supplied.

If the ig.dtdLookup property is removed or otherwise missing from the integrationGateway.properties file, the system responds as if the property is set to *True*, and request and response messages are validated against any associated DTD.

Setting Oracle Jolt Session Pooling Parameters

The integration gateway maintains a pool of jolt sessions to handle requests between itself and the integration engine. The integration gateway issues a jolt session from the pool, uses it for the connection, and then returns the session to the pool once it receives the response from the integration engine.

The number of sessions to maintain in the session pool is defined in the integrationGateway.properties file using the following property:

```
ig.connection
```

Set this property equal to the maximum number of sessions to maintain in the pool. The default value is *10*.

Setting the Namespace for Generic SOAP Faults

The system generates generic SOAP faults for framework-level errors, such as when it cannot find a routing for an integration.

To specify the namespace to use for generic SOAP faults, set the following property in the integrationGateway.properties file equal to the namespace to use:

```
ig.GenericFaultNamespace
```


Displaying the PeopleTools Version of Integration Gateways

You can enable the display of the PeopleTools version of an integration gateway as part of the information that displays when the gateway is pinged using the Ping Gateway button on the Gateways page.

Set the following property equal to *True* to enable the display of the PeopleTools version of a integration gateway:

`ig.Gateway.showDetails`

By default the property is set to *False* and the PeopleTools version of the gateway does not display when you ping the gateway.

Note: After you change the value for this property you must restart the web server for the new setting to take effect.

Masking Gateway Log File Elements

This section provides and overview of masking gateway log file elements and discusses how to:

- Access the logfilter.properties file.
- Mask elements not contained in namespaces.
- Mask elements contained in namespaces.
- Mask attributes of elements.
- Mask child element names.
- Change the global mask message.
- Create custom mask messages.
- Disable gateway log masks.

Understanding Masking Gateway Log File Elements

You can mask, or hide, elements that appear in the integration gateway log files, thereby prohibiting sensitive information from displaying in the generated logs.

Note: The system applies gateway log masks and messages to both the integration gateway message log file (MsgLog.html) and the integration gateway error log file (ErrorLog.html).

Global and Custom Mask Messages

By default, all masked elements have a global mask message applied to them, whereby every element you mask is replaced with a standardized message. You can also create custom mask messages for specific elements. You can use a combination of global and custom mask messages.

The default global mask message is *** *deleted for security purposes* ***. You can change the global mask as you wish to a message that best suits your business needs.

Default Masks

Several gateway log masks are implemented by default. They include, but are not limited to, the following elements:

- WSSE password.
- NodePassword.
- ExternalUserPassword.
- XML format request with password.
- PSFT AuthToken.
- SAML-TokenData.

You can disable any of these masks in the logfilter.properties file.

logfilter.properties File

To mask and unmask gateway log file elements use the logfilter.properties file.

To use the file to specify the element names, attribute names, and element namespaces to mask. You can also use the file to change the global mask message and set up custom mask messages.

Note: After you make any changes to the logfilter.properties file, you must reboot the web server for the changes to take effect.

Property Types

The following table lists the property types with which you can work in the logfilter.properties file:

Note: The examples provided in this section show property names appended with a number. These numbers are property indexes and are discussed elsewhere in this section.

Property	***Description***
AttributeName	Set this property equal to an attribute of an element to mask.
ElementName	Set this property equal to an element name to mask.
IsLeaf	Use this property to mask an element and all child tags of the element.
Namesapace	Use the Namespace property in conjunction with the ElementName property to specify the namespace of the element to mask.

Property Indexes

All properties in the logfilter.properties file are appended with an index number. Indexes group related properties and their values. The following example shows an excerpt from the logfilter.properties file and the ElementName.<index_number> naming scheme.

```
#IBInfo NodePassword
ElementName.2=NodePassword

#IBInfo ExternalUserPassword
ElementName.3=ExternalUserPassword

#XML format request with Password
ElementName.4=Password
```

You can use any number as an index number. Index numbers do not have to be used in sequence. Using the previous example, if you were to add a new element name to the file, you would not have to name it *ElementName.5.* You could use any number not already in use, such as *ElementName.72.*

Properties can appear in any order in the logfilter.properties file and do not have to appear in sequential index order. As an example, *ElementName.72* could appear first in the file, followed by *ElementName.3,* followed by *ElementName.1,* followed by *ElementName.12,* and so on.

Mask Variables

PeopleSoft Integration Broker provides the following mask variables:

Mask Variable	***Description***
GlobalReplaceWith	By default the system assigns the value of this variable to all asked elements. The default global mask message is: `***deleted for security purposes***`
ReplaceWith	Use this variable to override the global mask value for a specific element and set a custom mask message.

Accessing the logfilter.properties File

The logfilter.properties file is located in the following path in the PeopleSoft home directory:

```
<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\
WEB-INF\logfilter.properties
```

Masking Element Names Not Contained in Namespaces

Use the ElementName property to mask an element name that is not contained in a namespace.

To mask an element name that is not contained in a namespace, enter the element name to mask in logfilter.properties file in the following format:

```
ElementName.<index_number>=<Element_to_mask>
```


Be sure to specify a unique index number.

An example of a mask for an element name is shown in the following example.

```
ElementName.1=NodePassword
```

If you are using the default global mask, the element appears as follows in the gateway log files:

```
<NodePassword>*** deleted for security purposes ***</NodePassword>
```

Masking Element Names Contained Within Namespaces

Use the ElementName property and the Namespace property to mask elements contained within namespaces.

To mask an element name contain within a namespace, enter the element name to mask and namespace in which it is contained in the logfilter.properties file in the following format:

```
ElementName.<index_number>=<Element_to_mask>
Namespace.<index_number>=<Namespace_that_contains_element>
```

The ElementName and Namespace properties must use the same unique index number.

The following example shows how to enter a mask for the Username element contained in a namespace:

```
ElementName.9=Username
Namespace.9=http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd
```

If you are using the default global mask, the element appears as follows in the gateway log files:

```
<Username>*** deleted for security purposes ***</Username>
```

Masking Attributes of Element Names

Use the ElementName property and the AttributeName property to mask an attribute of an element.

To mask an attribute of an element, enter the element name and attribute name in the logfilter.properties file in the following format:

```
ElementName.<index_number>=<Element_name>
AttributeName.<index_number>=<Attribute_of_element_to_mask>
```

The ElementName and AttributeName properties must share the same unique index number.

The following example show the default mask for the password from the requesting node of a PeopleSoft 8.1x system:

```
#8.1x from node password
ElementName.6=from
AttributeName.6=password
```

An example request before masking is:

```
<?xml version="1.0"?>
<request version="1.0">
    <from node="PSFT_HR" password=password/>
    <to node="QE_LOCAL"/>
```


When the mask is applied, the request looks as follows:

```
<?xml version="1.0"?>
<request version="1.0">
    <from node="PSFT_HR" password=password/>
    <to node="QE_LOCAL"/>
```

Masking Child Element Names

Use and set the IsLeaf property equal to *false* to mask an element and all child elements. By default, child tags are not masked.

To mask an element and all child elements, enter the element name and set the IsLeaf property in the logfilter.properties file in the following format:

```
ElementName.<index_number>=<Element_name_(and_child_elements)_to_mask>
IsLeaf.<index_number>=false
```

The ElementName and IsLeaf properties must use the same unique index number.

As an example an address element could contain street number, street name, city, state, and zip code tags, as shown in the following example:

```
<address>
  <streetnumber>4433</streetnumber>
  <street>Oracle Lane</street>
  <city>Pleasanton</city>
  <state>California></state>
  <zipcode>94588</zipcode>
</address>
```

The following example shows how to mask the address element and all children of the element:

```
ElementName.11=address
IsLeaf.11=false
```

If you are using the default global mask, the element appears as follows in the gateway log files:

```
<address>***deleted for security purposes***</address>
```

However, if you wanted to mask just one of the child elements such as zip code, you would do so as shown in the following example:

```
ElementName.11=zipcode
```

The following example shows how the zip code tag would appear in the gateway logs if using the default global mask:

```
<address>
  <streetnumber>4433</streetnumber>
  <street>Oracle Lane</street>
  <city>Pleasanton</city>
  <state>California></state>
  <zipcode>***deleted for security purposes***</zipcode>
</address>
```


Changing the Global Mask Message

The value of the GlobalReplaceWith variable located in the logfilter.properties file determines the default global mask message. The default value is:

```
GlobalReplaceWith=***deleted for security purposes***
```

You can change this value as necessary to suit your business needs by setting the GlobalReplaceWith variable equal to another value. For example:

```
GlobalReplaceWith=####  PeopleSoft Confidential Information  ####
```

Creating Custom Mask Messages

You can override the global mask message on an element-by-element basis by setting the RepalceWith variable equal to a custom mask message.

The format is:

```
ElementName.<index.number>=<Element_to_mask>
ReplaceWith.<index_number>=<Custom_mask_message>
```

The index number you set must be the same unique index number used for the element, namespace, and/or attribute entry.

The following code snippet shows an example of overriding the default global mask message with a custom message:

```
#PSFT AuthToken
ElementName.7=AuthToken
ReplaceWith.7=-->Proprietary Information<--
```

When the gateway logs are generated the mask for this element will look as follows:

```
<AuthToken>-->Proprietary Information<--</AuthToken>
```

The following code example was shown earlier in this section. It has been modified to show how to override the default global mask message with a custom message:

```
ElementName.9=Username
Namespace.9=http://docs.oasis-open.org/wss/2004/01/
  oasis-200401-wss-wssecurity-secext-1.0.xsd
ReplaceWith.9=**  Data removed per company security policy **
```

When the gateway logs are generated the mask for this element will appear as follows:

```
<Username>**  Data removed per company security policy **</Username>
```

Disabling Gateway Log Masks

You can disable a mask for any element by commenting out the mask data in the logfilter.properties file.

For example, the following sample mask entry could appear in the logfilter.properties file:

```
#Sample Mask Entry
ElementName.44=NodeName
Namespace.44=http://my_namespace.xsd
ReplaceWith.44=--->Confidential/Proprietary Information<---
```

To disable the entry comment out all lines as shown in the following example:

```
#Sample Mask Entry
#ElementName.44=NodeName
#Namespace.44=http://my_namespace.xsd
#ReplaceWith.44=--->Confidential/Proprietary Information<---
```

Refreshing Integration Gateway Properties

If you modify integration gateway properties by accessing and directly modifying the integrationGateway.properties file located in the <PIA_HOME> directory, you must restart the web server for the changes to take effect.

If you modify integration gateway properties in the PeopleSoft Pure Internet Architecture, any changes you make take effect when you save the changes or click the OK button. This includes changes you make to the integrationGateway.properties file, but only if you access the file through the PeopleSoft Pure Internet Architecture using the Gateways Properties page.

Bypassing Integration Engines to Send Messages

You can use the PeopleCode built-in functions ConnectorRequest and ConnectorRequestURL to send synchronous requests directly to the integration gateway, without any message processing taking place on the integration broker engine, thereby eliminating the need for transactions.

Note: ConnectorRequest and ConnectorRequestURL are for use with synchronous requests only.

To use any of these methods, the integration gateway must be configured and running.

When you use either of these functions, errors and messages are written to the integration gateway logs.

Related Links

“Generating and Sending Messages” (Integration Broker)

Using the ConnectorRequest Built-In Function

The ConnectorRequest function enables you to build a message object and perform a POST or GET using any target connector. With this function, you use the Message object to populate connector values.

Response messages are returned unstructured in the *IB_GENERIC* message. The *IB_GENERIC* message is delivered out-of-the-box.

The ConnectorRequest function features optional user exception logic. If you pass the optional parameter *true* and a user exception occurs anywhere in the function, the response message returns a Response Status property set to *false.* Always read the Response Status property of the response message to determine if the call was successful. If it was not successful, interrogate the IBException object within the Message object to get the details of the exception.

The following example shows using the ConnectorRequest function to perform a GET to obtain a stock quote.

```
Local XmlDoc &Output;
Local String &Exception;

Local Message &MSG1, &MSG2;

&MSG = CreateMessage(OPERATION.QE_FLIGHTPLAN);

&MSG.IBInfo.IBConnectorInfo.ConnectorName = "HTTPTARGET";
&MSG.IBInfo.IBConnectorInfo.ConnectorClassName = "HttpTargetConnector";

&nReturn = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties
    ("Method", "GET", %HttpProperty);
&nReturn = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties
    ("URL", "http://finance.yahoo.com/d/quotes.txt/?symbols
    =PSFT&format=l1c1d1t1", %HttpProperty);

&MSG2 = %IntBroker.ConnectorRequest(&MSG, true); // user exception property (true) ⇒

passed

If &MSG2.ResponseStatus = %IB_Status_Success Then
    &Output = &MSG2.GetXmlDoc(); // get the data out of the message

Else
    &Exception = &MSG2.IBException.ToString()); // read the exception

End-If;
```

The following example shows sample code to read the response Message object to get exception details if using user exception logic:

```
&Response_MSG = %IntBroker.ConnectorRequest (&MSG, True);

If  &Response_MSG.ResponseStatus = %IB_Status_Success Then
    /* Perform successful processing of Response Message */

Else
    /* Read the IB Exception object for exception specifics. */
    &error = &Response_MSG.IBException.ToString ();

End-If;
```

Using the ConnectorRequestURL Built-In Function

The ConnectorRequestURL function enables you to use HTTP or FTP to perform a GET using a query string.

Based on the format of the string you provide, the integration gateway uses the HTTP target connector or FTP target connector to perform the GET.

Response messages are returned in a string.

Using ConnectorRequestURL with HTTP

The following example shows using the ConnectorRequestURL function to perform a GET to obtain a stock quote using HTTP.

```
&Output = %IntBroker.ConnectorRequestURL("http://finance.yahoo.com/d/quotes.txt/
?symbols=PSFT&format=l1c1d1t1");
```


Using ConnectorRequestURL with FTP

The syntax of the FTP URL is:

```
ftp://<user>:<password>@<host>:<port>/<url-path>;type=<typecode>
```

The following example shows using the ConnectorRequestURL function to perform a GET to obtain a stock quote using FTP.

```
&Output = %IntBroker.ConnectorRequestURL("ftp://qedmo:qedmo@ftp.globalsoft.com:
200/tmp/hello.xml;type=a");
```

Administering JMS Listening Connectors

This topic provides an overview of administering JMS listening connectors, describes prerequisites for administering JMS listening connectors, and discusses how to:

- Use the JMS Listening Connector Administration page.
- Start individual queue listeners and topic subscribers.
- Pause individual queue listeners and topic subscribers.
- Start all listener queues and topic subscribers.
- Stop all listener queues and topic subscribers.
- Auto-start queue listeners and topic subscribers.

Understanding Administering JMS Listening Connectors

The integration gateway provides access to a JMS Listening Connector Administration page that enables you view the number of available queue listeners and topic subscribers listening for JMS requests. In addition, you can use the page to perform administrative tasks such as:

- Start and pause individual listeners and subscribers.
- Start and stop all listeners and subscribers.

Note: The JMS Listening Connector Administration page is accessible from the local gateway only. This page will not work if an off-loaded URL is used for the gateway URL link.

Prerequisites for Administering JMS Listening Connectors

To administer JMS listening connectors:

- The JMS servlet must be started.
- The integration gateway URL must be defined.

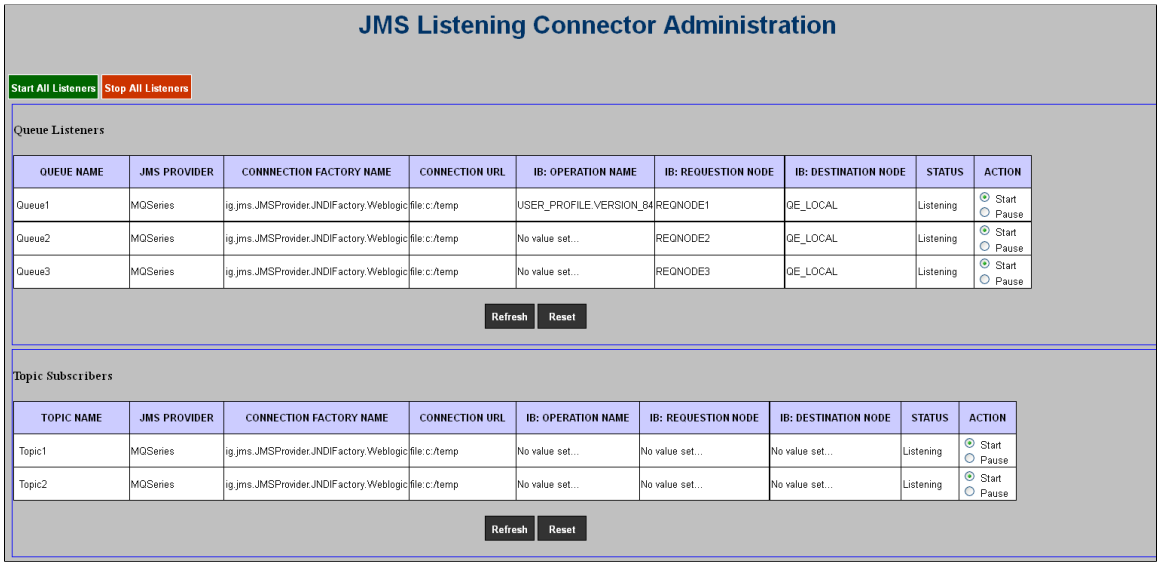
Note: The JMS Listening Connector Administration page is accessible from the local gateway only. This page will not work if an off-loaded URL is used for the gateway URL link.

- The integration gateway connectors must be loaded.
- JMS listening connectors must be configured in the integration gateway properties file, integrationGateway.properties.

Using the JMS Listening Connector Administration Page

To access JMS Listening Connector Administration page select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **JMS Administration** link. The page opens in a new browser window.

This example illustrates the fields and controls on the JMS Listening Connector Administration page. You can find definitions for the fields and controls later on this page.



The Queue Listener section of the page shows that three queue listeners are defined in the integration gateway. The Action column on the far right-side of the page shows that all three listeners are started and the Status column shows that all three queue listeners are listening

The Topic Subscribers section of the page shows that two topic subscribers are configured in the integration gateway. The Action column in the section shows that both subscribers are started and the Status column shows that both are in a listening status.

Note that in both sections of the page there are fields that display the message “No value set. .” This message appears in circumstances where optional gateway properties are not set for a particular property.

The page features the following fields and controls

Field or Control	***Description***
Start All Listeners	Click the button to start all listeners configured in the integration gateway properties file. If the action is successful, the **Status** field for all queue listeners and topic subscribers displays the status *Listening*.

Field or Control	***Description***
Stop All Listeners	Click the button to stop all listeners configured in the integration gateway properties file. If the action is successful the **Status** filed for all queue listeners and topic subscribers displays the status *Terminated.*
Queue Listeners (grid)	This grid displays fields and controls for JMS queue listeners. The fields and controls that appear in this grid are described within this table.
Topic Subscribers (grid)	This grid displays fields and controls for JMS topic subscribers. The fields and controls that appear in this grid are described within this table.
Refresh	Click the button to refresh all listeners or subscribers with changes made on the JMS Listening Connector Administration page and in the integration gateway properties file
Reset	Click the button to clear the form.
Queue Name	Name of the listening queue as defined in the integration gateway properties file.
Topic Name	Name of the subscription queue as defined in the integration gateway properties file.
JMS Provider	Name of the JMS provider.
Connection Factory Name	Name of the JNDIFactory class.
Connection URL	The JMS provider's URL to JNDI.
Operation Name	(Optional.) Specifies the name of the service operation and the service operation version.
Requesting Node	(Optional.) The name of the requesting node.
Destination Node	(Optional.) The name of the destination node.

Field or Control	*Description*
Status	Displays the status of the queue listener or topic subscriber. The possible values are: • *Listening.* The queue listener or topic subscriber is listening. • *Paused.* The queue listener or topic subscriber is paused. • *Terminated.* The queue listener or topic subscriber is stopped. • *Unknown.* This status displays when none of the other conditions apply, such as at start up.
Action	Click a radio button to start or pause a queue listener or topic subscriber as follows: • *Start.* Click this option to start a queue listener or topic subscriber. If the action is successful the **Status** field displays a status of *Listening.* • *Pause.* Click this option to pause a queue listener or topic subscriber. If the action is successful the **Status** field displays a status of *Paused.*

Fields denoted as *Optional* in the table correspond to optional JMS listening connector properties in the integration gateway properties file. If an optional field is not defined in the properties file no value appears for it on the JMS Listening Connector Administration page.

Starting Individual Queue Listeners and Topic Subscribers

To start individual queue listeners and topic subscribers:

1. Access the JMS Listening Connector Administration page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **JMS Administration** link.
2. Locate a queue listener or topic subscriber to start.
3. in the **Action** field click the **Start** control.
4. Click the **Refresh** button.

The **Status** field for the listener or subscriber displays the status of *Listening.*

It may take several moments for a listener or subscriber to start. If the status does not change immediately, click the **Refresh** button again.

Pausing Individual Queue Listeners and Topic Subscribers

To pause individual queue listeners and topic subscribers:

1. Access the JMS Listening Connector Administration page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **JMS Administration** link.

2. Locate a queue listener or topic subscriber to pause.
3. in the **Action** field click the **Pause** control.
4. Click the **Refresh** button.

The **Status** field for the listener or subscriber displays the status of *Paused.*

It may take several moments for a listener or subscriber to pause. If the status does not change immediately, click the **Refresh** button again.

Starting All Queue Listeners and Topic Subscribers

To start all queue listeners and all topic subscribers:

1. Access the JMS Listening Connector Administration page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **JMS Administration** link.
2. Click the **Start All Listeners** button.
3. Click the **Refresh** button.

The **Status** field for all listeners and subscribers displays the status of *Listening.*

It may take several moments for the listeners and subscribers to start. If the status does not change immediately, click the **Refresh** button again.

Stopping All Queue Listeners and Topic Subscribers

To stop all queue listeners and all topic subscribers

1. Access the JMS Listening Connector Administration page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **JMS Administration** link.
2. Click the **Stop All Listeners** button.
3. Click the **Refresh** button.

The **Status** field for all listeners and subscribers displays the status of *Terminated.*

It may take several moments for the listeners and subscribers to stop. If the status does not change immediately, click the **Refresh** button again.

Auto-Starting Queue Listeners and Topic Subscribers

You can set queue listeners and topic subscribers to start automatically during the boot up of the web server by adding the following value to the web.xml file for the PSIGW application:

```
<listener>
    <listenerclass>com.peoplesoft.pt.integrationgateway.common
   .JMSInitialize</listener-class>
</listener>
```

The location of the web.xml file is:

```
<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF
```


Chapter 5

Using Listening Connectors and Target Connectors

Understanding Listening Connectors and Target Connectors

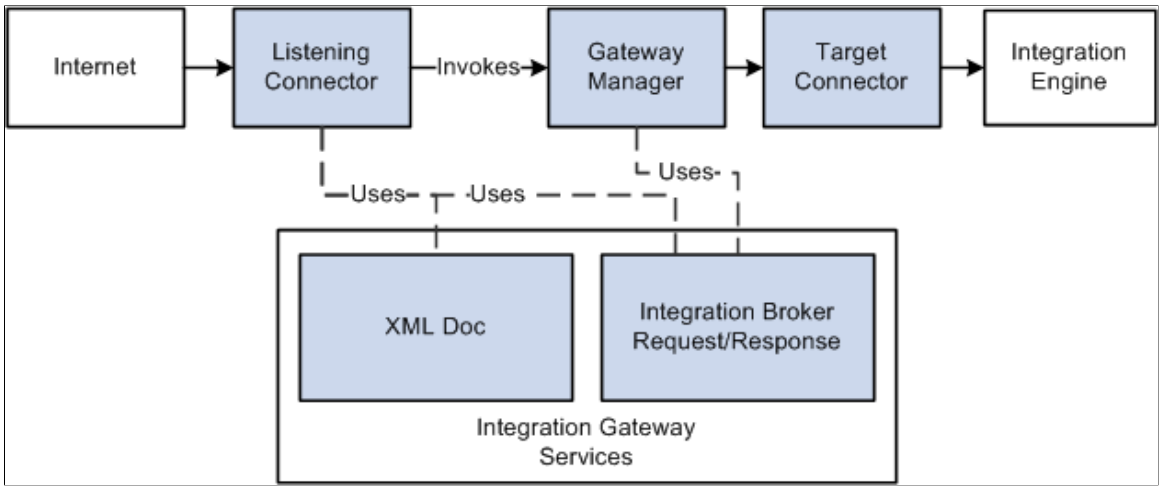
This topic discusses how to work with listening connectors and target connectors delivered with PeopleSoft Integration Broker. This topic also features code examples to help illustrate connector concepts and features.

Note: The code examples in this topic are for illustrative purposes only and are not intended to be used in a production environment.

Understanding Listening Connectors

Listening connectors receive requests from integration participants, send them to the gateway manager, and deliver responses back to the integration participants.

This example illustrates the flow of an inbound request from an external system into the integration engine through a listening connector.



PeopleSoft-Delivered Listening Connectors

PeopleSoft delivers several listening connectors with PeopleSoft Integration Broker that enable integration participants to communicate with the PeopleSoft system using a number of communication formats.

You send messages to a listening connector at a URL address derived from its class location on the gateway web server.

Note: The integration gateway provides a matching target connector for all connectors in the following table, except for the services listening connector. Although this topic discusses each pair of listening and target connectors in a separate section, the use of a particular listening connector does not obligate you to use the corresponding target connector.

Connector	*Description*
PeopleSoft listening connector	In combination with the PeopleSoft target connector, this connector establishes the primary connection between a PeopleSoft application's integration engine and its local gateway. It receives requests from integration participants in the PeopleSoft internal messaging format. Third-party applications and PeopleSoft releases that don't include PeopleSoft Integration Broker should not send messages to this connector. See Working With the PeopleSoft Connectors.
HTTP listening connector	This connector provides a web-standard method of communicating with the gateway. It accepts HTTP requests using the GET and POST methods. It also accepts secure HTTPS requests if SSL encryption is configured on the gateway's web server. See Working With the HTTP Connectors.
PeopleSoft 8.1 listening connector	This connector enables PeopleSoft 8.1x applications to communicate with the gateway using native Application Messaging technology. Third-party applications can send properly formatted 8.1x application messages to this connector. It also accepts secure HTTPS requests if SSL encryption is configured on the gateway's web server. See Working With the PeopleSoft 8.1 Connectors.
JMS listening connector	This connector enables JMS provider systems to communicate with the gateway using standard JMS protocols. See Working With the JMS Connectors.
AS2 listening connector	The AS2 listening connector enables you to receive request messages in AS2 format. See Working With the AS2 Connectors. **Note:** PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification. **Note:** AS2 Connectors will be desupported in future PeopleTools release.

Connector	*Description*
PeopleSoft services listening connector	PeopleSoft Integration Broker uses the PeopleSoftServiceListeningConnector as an endpoint for all node transactions that you expose as WSDL. All PeopleSoft node transactions that you publish as WSDL have the following endpoint: http://<machine>/PSIGW/PeopleSoftServiceListeningConnector. See Working With the PeopleSoft Services Listening Connector.
REST listening connector	The REST listening connector, RESTListeningConnector, accepts and processes REST based web services. PeopleSoft Integration Broker uses the RESTListeningConnector as an endpoint for all REST-based service operations defined in a WADL document. All service operations that you publish in a WADL have the following endpoint: http://<machine>/PSIGW/RESTListeningConnector. There is no configuration required for this connector and there are no properties to set or configure.

All of the delivered listening connectors that service HTTP requests run as servlets and are configured to run in the Oracle WebLogic web server environment. The delivered listening connectors that service HTTP requests are the PeopleSoft listening connector, the HTTP listening connector, and the PeopleSoft 8.1 listening connector.

Null Characters in Messages

The listening connectors delivered with PeopleSoft Integration Broker do not support null characters (ASCII value 00) as part of message field data. If a third-party application sends a message containing null characters, you must replace each instance of the null character with an acceptable substitute character, such as a space, before sending the message to the PeopleSoft system. Alternatively, you can modify the delivered listening connector to replace the null characters when it receives the message.

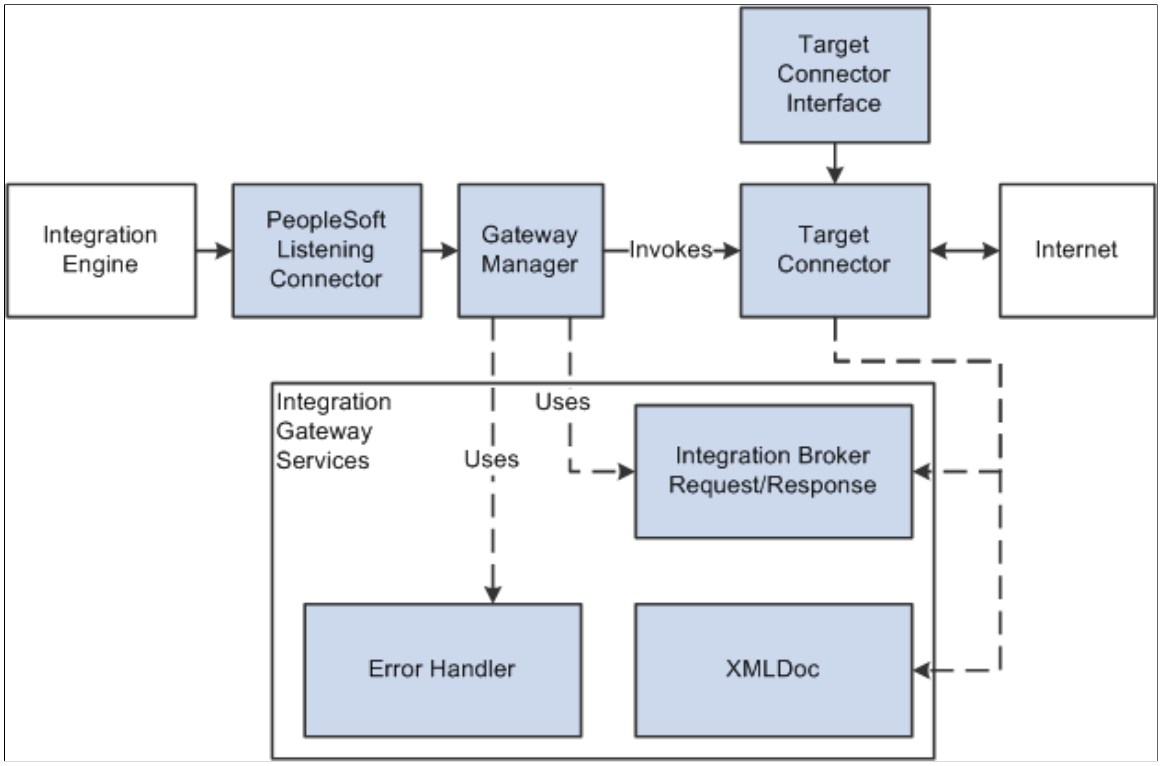
Listening Connectors and UTF Encoding

The listening connectors delivered with PeopleSoft Integration Broker support UTF-8 encoding.

Understanding Target Connectors

Target connectors generate requests, send them to integration participants, wait for responses from participants, and deliver the responses back to the gateway manager.

This example illustrates the request and response flow through a target connector.



The integration gateway invokes target connectors dynamically through the gateway manager. Target connectors adhere to a standard structure by implementing the target connector interface provided by the integration gateway. By implementing this interface, target connectors can take advantage of all gateway manager services.

Each target connector has an internal connector ID that you use when selecting the connector; for example, the connector ID for the HTTP target connector is *HTTPTARGET*.

PeopleSoft-Delivered Target Connectors

PeopleSoft delivers several target connectors with PeopleSoft Integration Broker that enable you to communicate with integration participants using a wide range of communication formats. The following table describes the delivered target connectors:

Connector ID	*Connector Class Name*	*Connector Name*	*Description*
AS2TARGET	AS2TargetConnector	AS2 target connector	The AS2 target connector enables you to send messages in AS2 format. See Working With the AS2 Connectors. **Note:** PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification. **Note:** AS2 Connectors will be desupported in future PeopleTools release.
FILEOUTPUT	SimpleFileTargetConnector	Simple file target connector	With this connector, the gateway saves outbound messages as XML files. **Note:** This connector (SimpleFileTargetConnector) is no longer a delivered connector and is now a part of the SDK. See Understanding the PeopleSoft Integration Broker Connector SDK.
FTPTARGET	FTPTargetConnector	FTP target connector	This connector enables the gateway to transfer messages to an FTP server. It converts outbound messages to file data it can send using the FTP PUT command. You can also send messages over a secure FTP (S) protocol. In addition you can receive messages from FTP servers using the GET command. See Working With the FTP Target Connector. **Note:** FTP Target Connector will be desupported in future PeopleTools release.
GETMAILTARGET	GetMailTargetConnector	GetMail target connector	This connector provides functionality specific to the PeopleSoft Multichannel Framework. See “Understanding the Email Channel” (MultiChannel Framework)

Connector ID	*Connector Class Name*	*Connector Name*	*Description*
HTTPTARGET	HttpTargetConnector	HTTP target connector	This connector provides a web-standard method for the gateway to communicate with PeopleSoft and third-party applications. It sends HTTP requests using the GET and POST methods. It also sends secure HTTPS requests if SSL encryption is configured on the gateway. See Working With the HTTP Connectors.
JMSTARGET	JMSTargetConnector	JMS target connector	This connector enables the gateway to communicate with JMS provider systems using standard JMS protocols. See Working With the JMS Connectors.
PSFT81TARGET	ApplicationMessagingTargetConnector	PeopleSoft 8.1 target connector	This connector enables the gateway to communicate with PeopleSoft 8.1x applications that use Application Messaging technology. It converts outbound messages to the Application Messaging native format. It also sends secure HTTPS requests if SSL encryption is configured on the gateway. See Working With the PeopleSoft 8.1 Connectors.
PSFTTARGET	PeopleSoftTargetConnector	PeopleSoft target connector	In combination with the PeopleSoft listening connector, this connector establishes the primary connection between a PeopleSoft application's integration engine and its local gateway. It sends requests to integration participants over a Oracle Jolt connection in the PeopleSoft internal messaging format. Use this connector to send messages only to PeopleSoft applications that use PeopleSoft Integration Broker. **Note:** Oracle Jolt is a Java-based interface that extends Oracle Tuxedo capabilities to the internet. The integration gateway uses it as the standard interface for communicating with integration engines through the PeopleSoft target connector. See Working With the PeopleSoft Connectors.

Connector ID	*Connector Class Name*	*Connector Name*	*Description*
SFTPTARGET	SFTPTargetConnector	SFTP target connector	The SFTP target connector enables the gateway to use SFTP to send messages to and receive messages from SFTP servers. It uses the PUT command to place messages or files from the integration gateway onto remote SFTP servers. The GET command is used to receive messages from SFTP servers. See Working With the SFTP Target Connector
SMTPTARGET	SMTPTargetConnector	SMTP target connector	With this connector, the gateway can send messages to an SMTP server using the PUT command. See Working With the SMTP Target Connector.

Target Connector Properties

Most of the delivered target connectors have required and optional configuration properties that you set to control the connectors' behavior. Depending on the connector, you configure some of these properties in the integrationGateway.properties file or by using the Gateways component. You can specify values for connector properties in the following ways:

- *Gateway-level* target connector properties always have the same value for a given connector, regardless of which nodes or transactions use the connector.

  You specify the values of these properties in the integrationGateway.properties file.

- *Node-level* target connector properties can have different values for each default local node that uses a given gateway.

  Each node-level connector property is identified by a property ID and a property name. You specify default values for these properties in the Gateways component of each participating node.

  See Administering Integration Gateways.

When you create a node definition in the local database, you specify which gateway and target connector should be used to send messages to that node. In the node definition, you can supply values for the connector's node-level properties that override the defaults and apply only when sending messages to that node.

See Specifying Gateways and Connectors.

When you define a routing definition, you can supply values for the connector's node-level properties to override the node definition's values and apply only when sending messages with that transaction.

See “Defining and Overriding Gateway and Connector Properties” (Integration Broker).

You can set and override target connector properties at runtime using PeopleCode.

See “Setting and Overriding Target Connector Properties at Runtime” (Integration Broker).

Target Connector Passwords

You must encrypt all required and optional target connector passwords.

See Encrypting Passwords.

Properties for HTTP URLs

The following connectors communicate over HTTP:

- AS2 target connector.

  Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

  Note: AS2 Connectors will be desupported in future PeopleTools release.

- HTTP target connector.
- PeopleSoft 8.1 target connector.
- PeopleSoft target connector.

For the HTTP target connector you can specify only one primary URL (PRIMARYURL) per node. The primary URL is the URL of the external system that handles the request.

However, you may specify more than one backup URL (BACKUPURL). Upon the failure of a transaction to the primary URL, the message is sent to any backup URLs one at a time. When a transaction that is sent to a URL succeeds, the other URLs are not used. If all URLs fail, the appropriate action and message is relayed to the calling module. The message and the node/URL failure is noted in the database or in the PeopleSoft Integration Broker Monitor.

Note: If the property ID is *HEADER,* then the target connector retrieves the information from a getHeader method call on the ConnectorInfo object, which resides on the IBRequest object. All other properties can be retrieved from a getFieldValue method call on the ConnectorInfo object.

Properties for Message Compression and Encoding

When the local integration gateway sends messages to a remote gateway, it ensures that they are compressed and base64 encoded. However, by default, when it sends messages directly to any node, it sends them uncompressed and unencoded. You can change this setting for transactions that use the following connectors:

- AS2 target connector.

  Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

  Note: AS2 Connectors will be desupported in future PeopleTools release.

- FTP target connector.

  Note: FTP Target Connector will be desupported in future PeopleTools release.

- HTTP target connector.
- JMS target connector.
- SMTP target connector.

Use the node-level SendUncompressed property for the appropriate connector. You can change the current value of this property specified for a given node by using the Connectors page of the node definition, or you can override the value for a single transaction by using the Connectors page of the node transaction detail. If you set the property's value to *No,* it sends messages compressed and base64 encoded.

See Specifying Gateways and Connectors.

Note: If nonrepudiation is in effect for a message, the SendUncompressed property is not used, and the message is always sent compressed and base64 encoded.

Setting Target Connector Delivery Modes

This section discusses how to:

- Specify the target connector delivery mode.
- Override the target connector delivery mode.
- Override the Service Operations Monitor contract status for Best Effort delivery transactions.

Understanding Setting Target Connector Delivery Modes

PeopleSoft provides the following delivery modes for asynchronous service operations:

Term	***Definition***
Guaranteed Delivery	If Integration Broker is not able to successfully deliver a service operation to its destination, the system automatically re-attempts delivery. The status of a service operation send using guaranteed delivery in the Service Operations Monitor is not *Done* until Integration Broker receives an acknowledgement from the receiving system. *Guaranteed Delivery* is the default delivery mode.

Term	*Definition*
Best Effort	The system makes one attempt to send a service operation to a destination. Upon sending, the transaction can appear in the Service Operations Monitor as *Done* or *Done NoAck.* A status of Done indicates that the transaction was sent to the target system and Integration Broker received an acknowledgement from the target system. A status of *Done NoAck* indicates that the transaction was sent, but Integration Broker did not receive an acknowledgement of the transaction from the receiving system. You can override the Service Operations Monitor contract status using PeopleCode.

Specifying Target Connector Delivery Modes

You can choose either *Guaranteed Delivery* or *Best Effort* delivery when using the following target connectors:

- AS2 target connector. (AS2TARGET.)

  Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

  Note: AS2 Connectors will be desupported in future PeopleTools release.

- File Output target connector. (FILEOUTPUT.)
- FTP target connector. (FTPTARGET.)

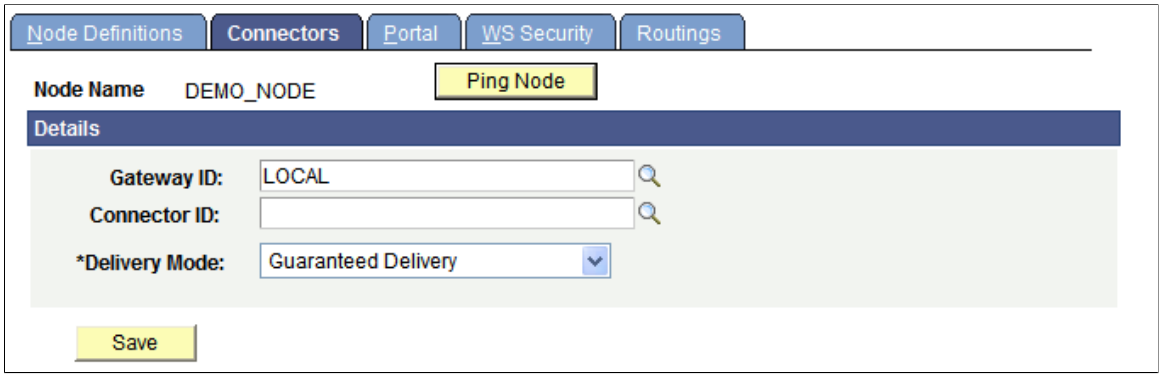
  Note: FTP Target Connector will be desupported in future PeopleTools release.

- SFTP target connector. (SFTPTARGET.)
- Get Mail target connector. (GETMAILTARGET)
- HTTP target connector. (HTTPTARGET.)
- JMS target connector. (JMSTARGET.)
- SMTP target connector. (SMPTTARGET.)

Guaranteed Delivery is the only delivery mode option available for the PeopleSoft 8.1 target connector (PSFT81TARGET) and PeopleSoft target connector (PSFTTARGET).

You set the delivery mode using the Connectors page (IB_NODECONN) in the Nodes component or the Connectors page (IB_ROUTINGDEFNCON) in the Routing component.

This example illustrates the Connectors tab with *Guaranteed Delivery* set as the delivery mode.



If working with the PeopleSoft 8.1 target connector or the PeopleSoft target connector, the Delivery Mode field is read-only.

Overriding Target Connector Delivery Modes

You can override the delivery mode set for a target connector by using the .IBConnectorInfo.DeliveryMode property of the IBInfo object and setting it equal to one of the following values:

Field or Control	***Description***
%IB_DM_BestEffort	Sets the delivery mode to *Best Effort.* Example: `&MSG.IBInfo.IBConnectorInfo.DeliveryMode⇒ = %IB_DM_BestEffort;`
%IB_DM_Guarantee	Sets the delivery mode to *Guaranteed Delivery.* Example: `&MSG.IBInfo.IBConnectorInfo.DeliveryMode⇒ = %IB_DM_Guarantee;`
%IB_DM_Reset	Clears any PeopleCode delivery mode override. Example: `&MSG.IBInfo.IBConnectorInfo.DeliveryMode⇒ = %IB_DM_Reset;`

The following pseudocode shows overriding the delivery mode to *Best Effort* prior to a publish:

```
&MSG=CreateMessage(OPERATION.FLIGHTPLAN);
&MSG.CopyRowset(&FLIGHT_PROFILE);
```



```
&Bo=&MSG.IBInfor.LoadConnectorPropFromRouting("FLIGHTPLAN_ASYNC");
&MSG.IBInfo.ConnectorOverride=True;
&MSG.IBInfo.ConnecotrInfo.DeliveryMode=%IB_DM_BestEffort;

%IntBroker.Publish(&MSG);
```

Overriding the Service Operations Monitor Contract Status for Best Effort Delivery Transactions

The system invokes the OnAckReceive PeopleCode event when the status of a publication contract is *Done,* or *Done NoAck.*. You can then override the contract status to the following: Done, Error, or Done NoAck.

In the case where best effort delivery was used, if the PeopleCode returns the status of Retry, the Integration Broker framework overrides the status to *Done NoAck.* To check if the status returned by the system is *Done NoAck* check the ResponseStatus property on the Message object.

The following pseudo-code demonstrates overriding the publication contract status of *Done NoAck* using the OnAckReceive event:

```
import PS_PT:Integration:IReceiver;

class FLIGHTACK implements PS_PT:Integration:IReceiver
   method FLIGHTACK();
   method OnAckReceive (&MSG As Message) Returns integer;
end-class;

/* constructor */
method FLIGHTACK
end-method;

method OnAckReceive
   /+ &MSG as Message +/
   /+ Returns Integer +/
   /+ Extends/implements PS_PT:Integration:Ireceive.OnAckReceive +/
   /+ Variable Declaration +/

   Local Rowset &rs;
   Local string &strException, &data;

   If &MSG.ResponseStatus <> %IB_Status_Success Then
     /+ Done NoAck status returned from IB framework
     can get the actual exception via the IBException Object +/
     &strException = &MSG.IBException.ToString();
     Return %Operation_DoneNoAck;
   End-If

   /* process soap fault and determine what to do +/
   &data = &MSG.GetContentString();

   Return %Operation_Done;

end-method
```

Working With the AS2 Connectors

This section discusses how to:

- Work with the AS2 listening connector.
- Work with AS2 response connector.

- Work with the AS2 target connector.

PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Note: AS2 Connectors will be desupported in future PeopleTools release.

Understanding Electronic Data Interchange Specifications Supported

Electronic Data Interchange (EDI) is a standard means of exchanging data between companies so that they can transact business electronically.

PeopleSoft supports the Applicability Statement 2 (AS2) specification for EDI. However, the Oracle SOA Suite B2B component supports AS2 and additional EDI formats and protocols, and provides a full-feature EDI integration solution.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

PeopleSoft provides generic integration capabilities with Oracle SOA B2B. Use Oracle Mediator-based services to integrate with Oracle SOA B2B.

Consult the Oracle SOA Suite documentation for EDI specifications supported.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Related Links

“Understanding Integrating with Oracle Mediator and Oracle ESB-Based Services” (Integration Broker)

Understanding Using AS2

AS2 is specification for Electronic Data Interchange (EDI) between organizations using the internet. AS2 uses Secure/Multipurpose Internet Mail Extensions (S/MIME), which secures data with authentication, nonrepudiation and encryption. The transportation protocol for this specification is HTTP and HTTPS for real-time communication. S/MIME secures data with authentication, message integrity and nonrepudiation.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

PeopleSoft Integration Broker provides three connectors for use with AS2:

Term	***Definition***
AS2 listening connector	Use the AS2 listening connector to receive request messages in AS2 format.

Term	*Definition*
AS2 response connector	The AS2 response connector sends acknowledgements for data you receive from the AS2 listening connector.
AS2 target connector	Use the AS2 target connector to send messages in AS2 format.

You can use the AS2 listening and target connectors to transport any kind of data, including, but not limited to, XML, EDI, text and binary data.

The AS2 target connect is segment-aware and you may use it to send message segments to integration partners.

See “Working With Message Segments” (Integration Broker).

Understanding MDNs

AS2 uses two different message types: the request message containing the data to be integrated and the Message Disposition Notification (MDN) to acknowledge the receipt of the data.

AS2 message exchange can occur over HTTP or HTTPS. The sender must request and MDN from the receiver, that enables the sender to verify that the message has been transferred in an unmodified state and that the receiver has been able to decompress or decrypt the message.

As an option, an MDN may be digitally signed, enabling the recipient to authenticate the sender of the MDN to check the integrity of the incoming message.

MDNs can be delivered synchronously or asynchronously.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Synchronous MDNs

Synchronous MDNs are returned to the sender in the same HTTP connection that sent the message. Processing does not continue until the sender receives the MDN.

Asynchronous MDNs

Asynchronous MDNs are delivered to the sender at a later time after the transmission of the message.

AS2 Requests initiated by the AS2 target connector with an asynchronous MDN Type must send MDN asynchronous responses to the AS2 response connector at the following URL:

```
http://<SERVER><PORT>/PSIGW/AS2ResponseConnector
```

The AS2 response connector processes MDNs by verifying them with sent request and publishes a response message to the PeopleSoft Integration Broker.

When a message is published the AS2 target connector stores the information regarding the request (for example, Message ID, signed algorithm, and so forth) for verifying the response on the integration

gateway. When the response is received, the AS2 response connector verifies with the request information and publishes a response message to PeopleSoft Integration Broker.

A published asynchronous response is an empty message with the following structure:

```
<? Xml version="1.0">
<AS2ASyncResponse>
      <ConversationID>123213</ConversationID>
      <OriginalMessageID>23234<OriginalMessageID>
      <MDN>123123 . . </MDN>
      <ReceiptMessage> <![CDATA[Receipt message.]]></ReceiptMessage>
      <MDNVerified>True/False<MDNVerified>
</AS2ASyncResponse>
```

PeopleSoft Integration Broker generates the conversation ID tag when a message is published. This tag is used to correlate the MDN with the request message.

If the MDNVerified tag is set to *True*, the integration gateway has successfully verified the MDN.

Note: To provide application the flexibility to take appropriate action with responses and response status information, it is the developer's responsibility to write subscription PeopleCode for processing acknowledgement messages and correlating them with requests. Without subscription PeopleCode to consume the message, an MDN will not be sent back to the source.

The AS2 connectors implement correlation IDs in MDNs. The AS2 target connector saves the outbound message ID as a correlation ID in the directory defined in the ig.AS2.AS2Directory in the integrationGateway.properties file .

When the response arrives later, the AS2ResponseConnector checks the conversationID from the response message with the one saved by early. If they don’t match, the transaction fails.

PeopleCode Considerations

In outbound messages, always use the %Intbroker.publish () function. Using %IntBroker.SyncRequest results in errors.

Understanding the AS2 Listening Connector

The AS2 listening connector can receive inbound asynchronous request messages, and can send synchronous and asynchronous MDNs. This section describes how these messages flow through the AS2 listening connector and how MDNs are created and returned to the senders of messages.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Inbound Asynchronous Request—Synchronous MDN

This section describes the process flow of an inbound asynchronous request message through the AS2 listening connector, with the integration engine generating a synchronous MDN.

1. The AS2 listening connector receives an AS2 message over an open HTTP connection.
2. The connector verifies the digital signature and decrypts the message. If necessary, the connector also decompresses the message.

3. The AS2 listening connector sends the message to the integration engine.
4. The integration engine creates an MDN and sends it back to the integration gateway as part of the HTTP response message.

Inbound asynchronous Request—Asynchronous MDN

This section describes the process flow of an inbound asynchronous request message through the AS2 listening connector, with the integration engine generating an asynchronous MDN.

1. The AS2 listening connector receives a message over HTTP.
2. The AS2 listening connector closes the connection and sends a status code of *200.*
3. The connector verifies the digital signature and decrypts the message. If necessary, the connector also decompresses the message.
4. The AS2 listening connector sends the message to the integration engine.
5. The integration engine creates an MDN and sends it back to the sender as an asynchronous transaction, using the AS2 target connector.

Understanding the AS2 Response Connector

When a request is published, PeopleSoft Integration Broker generates a conversation ID in the message ID field of the request message. Then, when an MDN comes back it extracts the conversation ID from the message to correlate the MDN acknowledgement with the request message.

Note: You must write subscription PeopleCode to process acknowledgement messages and to correlate them with requests messages. This provides flexibility for you to specify actions to take based on response status.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

When it receives an MDN, the AS2 response connector checks for the conversation ID, constructs the asynchronous response message by setting the conversation ID, MDN, and the message/subject received with the MDN. It then sends the response to the integration engine.

Understanding the AS2 Target Connector

This section describes how messages flow through the AS2 target connector and how the connector processes MDNs.

Note: The AS2 target connector sends message requests in asynchronous mode only. However, the connector can receive MDNs in synchronous or asynchronous mode.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Outbound Asynchronous Request—Synchronous MDN

This section describes the process flow of outbound asynchronous request message through the AS2 listening connector, with the integration engine generating a synchronous MDN.

1. The AS2 target connector receives the request message from the integration engine.
2. The AS2 target connector checks the outbound message to determine if an MDN is required, and if so, whether the MDN is synchronous or asynchronous.
3. The AS2 connector makes an HTTP request to the receiver.
4. The AS2 connector verifies the MDN in the HTTP response if an MDN is requested.
5. Once the MDN is verified, the AS2 connector sends a response to the integration engine indicating whether the message was sent successfully.

Outbound Asynchronous Request—Asynchronous MDN

This section describes the process flow of an outbound synchronous request message through the AS2 listening connector, with the integration engine generating an asynchronous MDN.

1. The AS2 target connector receives the request message from the integration engine.
2. The AS2 target connector checks the outbound message to determine if an MDN is required, and if so, whether the MDN is synchronous or asynchronous.
3. Check for MDNAsynchronousURL and request a Asynchronous Receipt (MDN).
4. The AS2 connector makes an HTTP request to the receiver.
5. The AS2 connector reads the HTTP status code and sends a response to the integration engine indicating whether the message was sent successfully.
6. At a later time, the AS2 listening connector receives an MDN from the receiver. The MDN is then processed.

   See Understanding MDNs.

Using the AS2 Listening Connector

This section describes how to use the AS2 listening connector and discusses how to:

- Set required header parameters.
- Set optional header parameters.
- Set gateway-level properties.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Setting Required Header Parameters

The following HTTP header parameters are required in incoming AS2 requests:

HTTP Header Parameter	*Description*
AS2From	Specify the name of the sending node.
AS2To	Specify the name of the receiving node.
MessageName	Specify the name of the incoming operation or message. **Note:** You can specify the message name in the HTTP header, HTTP query string or in the integrationGateway.properties file. The value is in the form Message.Version. For example: *UserProfile.Version_84*

If the AS2From and AS2To node names are not PeopleSoft node names, you must map them in the integrationGateway.properties file.

Setting Optional Header Parameters

When using the AS2 listening connector, you may set the following optional HTTP header parameters:

HTTP Header Parameter	*Description*
Password	(Optional.) Specify an encrypted password for node authentication.
OrigUser	(Optional.) Specify the user name of the originating user.
ExternalMessageID	(Optional.) Specify a unique ID that identifies the message. If two messages are published with the same external message ID, the first message is processed and the second messages is marked as a duplicate.

Setting Gateway-Level Properties

To configure the AS2 listening connector, you must set properties located in the AS2 Connector Properties of the integrationGateway.properties file for each message the connector receives.

Note: Replace text in angle brackets (for example <*project_branch*>) with the appropriate values.

A property is required unless denoted as *"(Optional.)"* in the description.

Property	*Description*
ig.AS2.LogDirectory	(Optional.) Specify the directory to log all incoming and outgoing AS2 requests and responses. For example: *ig.AS2.LogDirectory = c://temp//as2//logs*
ig.AS2.KeyStorePath	Specify the path to the Java keystore. For example: *C://pt853 //webserv//peoplesoft//keystore//pskey*
ig.AS2.KeyStorePassword	Specify the encrypted password to the Java keystore.
ig.AS2.AS2ListenerMap.From.<from alias>	(Optional.) If a sending or receiving node is not a PeopleSoft node, you must map it in the integrationGateway.properties file. Use this property if the sending system is not a PeopleSoft node. Replace the information in brackets with an alias of the sending system and set it equal to the remote node name in the PeopleSoft application database. For example: *ig.AS2.AS2ListenerMap.From.QE_SOURCE= PT_LOCAL*
ig.AS2.AS2ListenerMap.To.<to alias>	(Optional.) If a sending or receiving node is not a PeopleSoft node, you must map it in the integrationGateway.properties file. Use this property if the receiving system is not a PeopleSoft node. Replace the information in brackets with an alias of the receiving system and set it equal to the remote node name in the PeopleSoft application database. For example: *ig.AS2.AS2ListenerMap.To. QE_IBTGT= AS2TARGETNODE*

Property	*Description*
ig.AS2.<source>.<target>.CertificateAlias	Specify the certificate (target) alias name. Replace <source> and <target> with the source and target PeopleSoft node names used in the AS2FROM and AS2TO HTTP headers, or those mapped in the properties above. For example: *ig.AS2.PT_LOCAL.AS2TARGETNODE. CertificateAlias=JFRANCO030204*
ig.AS2.<source>.<target>.SignerCertificateAlias	Specify the certificate alias (source) used for signing the certificate. For example: *ig.AS2.PT_LOCAL.AS2TARGETNODE. SignerCertificateAlias=JRICHAR2030104*
ig.AS2.<source>.<target>.MessageName	(Optional.) Specify the name of the incoming message. Replace <source> and <target> with the source and target PeopleSoft node names used in the AS2FROM and AS2TO HTTP headers, or those mapped in the properties above. For example: *ig.AS2. PT_LOCAL.AS2TARGETNODE. MessageName=EXAMPLE_REQUEST_MSG* **Note:** You can specify the message name in the HTTP header, HTTP query string or in the integrationGateway.properties file.

Using the AS2 Target Connector

This section describes using the AS2 target connector and discusses how to:

- Set node-level connector properties.
- Set gateway-level connector properties.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Setting Node-Level Connector Properties

The following table lists the required and optional AS2 target connector properties you set at the node level. You set these properties in the Gateways component in the PeopleSoft Pure Internet Architecture.

A property is required unless denoted as *"(Optional.)"* in the description.

Property ID	***Property***	***Description***
AS2PROPERTY	AS2From	Specify the name of the sending node.
AS2PROPERTY	AS2To	Specify the name of the receiving node.
AS2PROPERTY	AsynchronousMDNRecipientURL	Specify a URL that indicates how and where the MDN is delivered. For example: *http://<source webserver>:<http port>/PSIGW/AS2ResponseConnector* By specifying a valid URL you can request asynchronous delivery instead. The URL indicates the destination for the reply, and may use any appropriate protocol, such as HTTP or HTTPS. If this property is set to an empty string (Default), the receipt is returned synchronously within an HTTP reply.
AS2PROPERTY	Compression	Specify whether to compress outbound AS2 messages. Options are: • *Y*: Send messages compressed using the Zlib compression format. • *N*: No compression. (Default.)
AS2PROPERTY	EDIType	Specify the content type of the message. Options are: • *Application/edi-x12.* • *Application/edifact.* • *Application/xml..* • *Application/text.*

Property ID	*Property*	*Description*
AS2PROPERTY	EnableCRLF	(Optional.) PeopleSoft Integration Broker automatically removes carriage returns in messages and retains line feeds. Use this property to specify whether to add a carriage return (CR) back to the end of a line feed (LF). Options are: • *Y*. Adds CR to LF. (Default.) • *N*. No CR added to LF.
AS2PROPERTY	EncryptingAlgorithm	(Optional.) Specify the algorithm used to encrypt data. The default value is *3DES*. Use of this algorithm is highly recommended. When you specify an encrypting algorithm, you must set the RecipientCertAlias to a valid certificate. The data is encrypted using the RecipientCertAlias value you define with the algorithm you specify here.
AS2PROPERTY	FirewallHost	(Optional.) If connecting through a firewall, specify the firewall host name or IP address.
AS2PROPERTY	FirewallPassword	(Optional.) If connecting through a firewall, specify an encrypted password if authentication is to be used when connecting through the firewall.
AS2PROPERTY	FirewallPort	(Optional.) If connecting through a firewall, specify the port of the firewall to which to connect. See the description for the FirewallType property for guidelines on how the default setting is made.

Property ID	*Property*	*Description*
AS2PROPERTY	FirewallType	(Optional.) If connecting through a firewall, specify the type of firewall. Options are: • *NoFirewall*. (Default.) • *TunnelingProxy*: Connects through a tunneling proxy. The FirewallPort property is automatically set to *80*. • *SOCK4Proxy*: Connects through a SOCKS4 proxy. The FirewallPort property is automatically set to *1080*. • *SOCK5Proxy*Connects through a SOCKS5 proxy. The FirewallPort property is automatically set to *1080* You can overwrite port numbers in the FirewallProperty field.
AS2PROPERTY	Firewall User	(Optional.) If connecting through a firewall, specify the firewall user name if authentication is to be used connecting through a firewall.
AS2PROPERTY	Http Password	(Optional.) Specify the HTTP user name if HTTP authentication is to be used.
AS2PROPERTY	HttpUser	(Optional.) Specify the HTTP user name password if HTTP authentication is to be used.
AS2PROPERTY	MDNSecurityType	(Optional.) Specify the algorithm to use for signing the MDN. Options are: • *Signed-sha1*. (Default.) • *Signed-md5*. • *Unsigned*.

Property ID	*Property*	*Description*
AS2PROPERTY	MDNType	Specify whether to generate an MDN, and if so the type to generate. Options are: • *None:* • *Sync:* Synchronous. (Default.) • *Async:* Asynchronous.
AS2PROPERTY	ProxyPassword	(Optional.) Specify the proxy user password.
AS2PROPERTY	ProxyPort	(Optional.) Port of the proxy server to which to connect.
AS2PROPERTY	ProxySSL	(Optional.) Options are: • *Automatic*. (Default.) • *Always*. • *Never*. • *Tunnel*.
AS2PROPERTY	ProxyServer	(Optional.) Specify the proxy server name or IP address.
AS2PROPERTY	ProxyUser	(Optional.) Specify the user name if authentication is to be used to connect through a proxy.
AS2PROPERTY	RecipientCertAlias	(Optional.) Specify the alias name of the recipient's certificate. **Note:** This property is required if the EncryptingAlgorithm property is set.

Property ID	***Property***	***Description***
AS2PROPERTY	SecurityType	Specify the security type of the request message. Options are: • *EncryptedOnly*. • *Signed-Encrypted*. (Default.) • *SignedOnly*. • *None*.
AS2PROPERTY	SignersCertificateSubject	Specify the alias name of the signing certificate. This property is required if the SecurityType property is set to *SignedOnly* or *Signed-Encrypted*.
AS2PROPERTY	TimeOut	(Optional.) Specify the timeout for the connector in seconds. When this value is set to *0*, all operation will run uninterrupted until successful completion, or an error condition is encountered. The default value is *60*.
AS2PROPERTY	User Agent	(Optional.) Specify the name of the user agent or email address.
BACKUPURL	URL	(Optional.) Specify the backup URL to use to send messages if delivery to the primary URL fails.
PRIMARYURL	URL	Specify the URL to which messages are sent using this connector.

Property ID	*Property*	*Description*
HEADER	sendUncompressed	Specify whether to send messages decompressed. Options are: *Y:* Send messages decompressed and decoded. (Default.) *N:* Send messages compressed and base64 encoded. **Note:** Do not change the default value.
PRIMARYURL	URL	Specify the URL to which messages are sent using this connector. For example: *http://<target webserver>:<http port>/PSIGW/AS2ListeningConnector*

Setting Gateway-Level Connector Properties

This section describes required AS2 target connector properties you set in the integrationGateway.properties file.

A property is required unless denoted as *"(Optional.)"* in the description.

The AS2 target connector uses digital certificates for digital signatures, nonrepudiation and encryption.

As a result, you must set up digital certificates to use the connector.

Public keys and signatures are stored in certificates, so there must be a place in the organization to store these keys and certificates.

The place to store keys is the key store. A key store can be a flat file, a database or an LDAP server that can store key material. PeopleSoft keystore is installed with the PeopleSoft Pure Internet Architecture at the following default location: <PIA_HOME>\webserver\<DOMAIN>\keystore. PeopleSoft AS2 connectors will invoke these certificates from JKS. JKS exists on the web server.

The following properties should be set in the AS2 Connector Properties section in the integrationGateway.properties file of the source web server in order to use the AS2 target connector. Use the PSCipher utility to encrypt the password.

Property	*Description*
ig.AS2.KeyStorePath	Specify the path to the Java keystore. For example: *C://pt854//webserv//peoplesoft//keystore//pskey*
ig.AS2.KeyStorePassword	Specify the encrypted password to the Java keystore.
ig.AS2.AS2Directory	Specify the directory to log MDN responses. This property is required for asynchronous MDNs. For example: *c://temp//as2*
ig.AS2.LogDirectory	(Optional.) Specify the directory to log all incoming and outgoing AS2 requests and responses. For example: *c://temp//as2//logs*

Working With the FTP Target Connector

This section discusses working with the FTP target connector.

Note: FTP Target Connector will be desupported in future PeopleTools release.

Understanding the FTP Target Connector

The FTP target connector enables the gateway to use FTP to send messages to and receive messages from FTP servers. It uses the PUT command to place messages or files from the integration gateway onto remote FTP servers. The GET command is used to receive messages from FTP servers. Outbound messages through the FTP target connector are UTF-8 encoded.

PeopleSoft Integration Broker also supports secure communication with FTP servers using FTPS.

Note: The FTP target connector handles string-based data only. Binary data is not natively supported in PeopleSoft Integration Broker.

The connector ID for the FTP target connector is *FTPTARGET*.

Prerequisites for Using the FTP Target Connector

In addition to specifying Java Archive (JAR) files in the web server CLASSPATH and setting node-level connector properties, to use this connector you must also specify the integration gateway URL in the Gateways component.

Information about specifying the required JAR files and setting node-level FTP and FTPS connector properties is discussed in this section.

See Specifying Required JAR Files, Setting Node-Level FTP Connector Properties, Setting Node-Level FTPS Connector Properties.

Information about specifying the integration gateway URL is discussed elsewhere in the product documentation.

See Administering Integration Gateways.

If using an IIS FTP server with the FTP target connector, ensure the directory listing style in IIS is configured with type as UNIX and not as MS-DOS.

Specifying Required JAR Files

For the FTP target connector to function properly the following JAR files from IBM must reside in the CLASSPATH of the web server running the integration gateway:

- FTPProtocol.jar
- ipworksssl.jar (required for FTPS)

Setting Node-Level FTP Connector Properties

This section describes the required node-level properties you must set to use the FTP target connector.

The following table describes the required node-level connector properties:

Property ID	***Property Name***	***Description***
HEADER	SendUncompressed	Specify whether to send messages decompressed. Values are: • *Y:* Send messages decompressed and decoded. This is the default value. • *N:* Send messages compressed and base64 encoded.

Property ID	*Property Name*	*Description*
FTPTARGET	FTPMODE	Specify whether to use an active or passive FTP connection for integrations. The valid values are: • *ACTIVE.* (Default.) Use an active FTP connection. • *PASSIVE* Use a passive FTP connection.
FTPTARGET	HOSTNAME	Specify the IP address or name of the FTP server for the connection.
FTPTARGET	METHOD	Specify the method to send or receive messages. The valid values are: • *PUT* (default). Send messages to an FTP server. • *GET.* Retrieve messages from an FTP server. • *GETDIRLIST.* Retrieve a directory list of files from an FTP server.
FTPTARGET	DIRECTORY	Specify the remote directory into which the file is placed. **Note:** When using the GET method you must specify the location where the file resides for the method to function properly. If not specified, the default directory of the FTP server on the remote site is used.

Property ID	*Property Name*	*Description*
FTPTARGET	FILENAME	(Optional.) Specify the name of the file saved on the recipient's FTP server. By default, the file name is a concatenation of the following: • Originating node name. • Originating user name. • Operation name. • Originating timestamp. • Segment ID. If you do not specify a filename, the FTP (S) target connector performs a GET to retrieve the directory list from the remote FTP server. See the section on Directory List Support earlier in this section.
FTPTARGET	USERNAME	Enter the FTP server login ID.
FTPTARGET	PASSWORD	Enter the password for the login to the FTP server. This password must be encrypted. See Encrypting Passwords.
FTPTARGET	TIMEOUT	Specify the time in milliseconds for the connector to wait for the message to transmit. If the timeout period expires without a successful transmission, the transaction fails. The default value is *50000* (50 seconds).
FTPTARGET	TYPE	Indicates the FTP mode used to transfer the file. The valid options are: • *ASCII* (default) • *BINARY* When you select *ASCII*, all characters are converted to their ASCII equivalents. When you select *BINARY*, data is copied bit-by-bit and no conversion is performed.

Setting Node-Level FTPS Connector Properties

The following table describes properties to use for secure FTPS communication.

Property ID	*Property Name*	*Description*
FTPTARGET	FTPS	Enables secure communication over FTP. Values are: • *Y:* Enable FTPS communication. • *N:* Disable FTPS communication. This is the default value.
FTPTARGET	CLIENTCERT	(Optional.) To use client authentication when establishing a connection with the target or receiving system, enable the CLIENTCERT property.
FTPTARGET	PORT	Specify the port used for communication. The default port is *21*.
FTPTARGET	SSLSTARTMODE	(Optional). Use this property to set the SSL start mode. Values are: • *DEFAULT.* If the remote port is set to the standard plain text port of the protocol (where applicable), it will behave the same as if SSLSTARTMODE is set to *sslExplicit* . In all other cases, SSL negotiation will be implicit (*sslImplicit*). • *IMPLICIT.* The SSL negotiation will start immediately after the connection is established. • *EXPLICIT.* The connector first connects in plain text, and then explicitly starts SSL negotiation through a protocol command such as STARTTLS.

Using Directory Lists

One of the optional node-level FTP connector properties is FILENAME. If you do not know the file name of the file you would like to receive but do not know the directory in which it resides, you can use the GETDIRLIST method to retrieve a directory list. The directory list is retrieved in XML format and you must parse the XML document to read its contents. You can then use the GET method to get the actual file. The following example shows the format of a returned directory list.

```
<DirList>
   <File name="sample.bat">
      <Date></Date>
```



```
      <Size>1234</Size>
      <Time></Time>
      <isFile>True</isFile>
   </File>
   <File name="sample2.bat">
      <Date></Date>
      <Size>1234</Size>
      <Time></Time>
      <isFile>True</isFile>
   </File>
   <File name="temp">
      <Date></Date>
      <Size>1234</Size>
      <Time></Time>
      <isFile>False</isFile>
   </File>
</DirList>


Date : Date on the file on remote system
Time : Time on the file on remote system
Size : Size of the file
isFile : True if it is a file. False if it is a directory.
```

Directory List Example

The following example shows the code needed to use the FTP connector to get a list of the files in a directory, run through the list of files, select a file, and retrieve it. To use this example, you must know the directory in which the file resides.

If you know the name of the file you wish to receive but do not know the directory, use the FILENAME property and the GETDIRLIST method to retrieve a directory list, as described previously in this section.

See Using Directory Lists.

```
Local XmlDoc &Output;
Local Message &MSG1, &MSG2, &MSG3;

&MSG = CreateMessage(OPERATION.QE_FLIGHTPLAN_UNSTRUCT);

/* Set ConnectorName and Connector ClassName */
&MSG.IBInfo.IBConnectorInfo.ConnectorName = "FTPTARGET";
&MSG.IBInfo.IBConnectorInfo.ConnectorClassName = "FTPTargetConnector";

/* Set the FTP connector properties in the ConnectorInfo */
/* Method name can be either Get or GetDirlist.  */
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("METHOD",
     "GET",%Property);
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("HOSTNAME",
     "ftp.example.com",%Property);
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("USERNAME",
      "sam",%Property);

/*  Encrypt the password */
&pscipher = CreateJavaObject("com.peoplesoft.pt.integrationgateway.common.
     EncryptPassword");
&encPassword= &pscipher.encryptPassword("ftpserverpassword");
 &pscipher = Null;

&string_return_value = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties
     ("PASSWORD", encPassword,%Property,);
&string_return_value = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties
     ("DIRECTORY", "/incoming/tmp",);

/* Do Connector Request */
&MSG2 = %IntBroker.ConnectorRequest(&MSG);
```



```
/* Get XMLDoc from MSG2*/
&fileListXmlDoc = &MSG2.GetXmlDoc();

/*Parse the XMLDoc. Structure of the DirList Message is
<DirList>
     <File name="sample.bat">
        <Date></Date>
        <Size>1234</Size>
        <Time></Time>
        <isFile>True/False</isFile>
     </File>
</DirList>*/

&XmlNode = &fileListXmlDoc.DocumentElement.FindNode("/DirList/File");

/* Get the file name */
&attName = &XmlNode.GetAttributeName(1);
&fileName = &XmlNode.GetAttributeValue(&attName);

/* Get the file name from the Remote FTPServer */
&MSG = CreateMessage(OPERATION.QE_FLIGHTPLAN_UNSTRUCT);

/* Set ConnectorName and Connector ClassName */
&MSG.IBInfo.IBConnectorInfo.ConnectorName = "FTPTARGET";
&MSG.IBInfo.IBConnectorInfo.ConnectorClassName = "FTPTargetConnector";

/* Set the FTP connector properties in the ConnectorInfo */
/* Mehtod name can be either Get */
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("METHOD",
   "GET",%Property);
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("FILENAME",
     &fileName,%Property);
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("HOSTNAME",
     "ftp.example.com",%Property);
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("USERNAME",
   "sam",%Property);

/*  Encrypt the password */
&pscipher = CreateJavaObject("com.peoplesoft.pt.integrationgateway.common.
     EncryptPassword");
&encPassword= &pscipher.encryptPassword("ftpserverpassword");
 &pscipher = Null;

&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("PASSWORD",
     encPassword,%Property,);
&nRet = &MSG.IBinfo.IBConnectorInfo.AddConnectorProperties("DIRECTORY",
   "/incoming/tmp",);

/* Do Connector Request */
&MSG3 = %IntBroker.ConnectorRequest(&MSG);
```

Working With the HTTP Connectors

This section provides an overview of the HTTP connectors and discusses how to:

- Use the HTTP listening connector.
- Use the HTTP target connector.
- Comply with message formatting and transmission requirements.
- Run the gateway behind a proxy server.

Understanding the HTTP Connectors

The HTTP listening and target connectors provide a web-standard method for an integration gateway to exchange messages with both PeopleSoft and third-party applications using the HTTP GET and POST methods. They also support secure HTTPS communications if SSL encryption is configured on the gateway machine.

Using the HTTP Listening Connector

The HTTP listening connector monitors specific ports for incoming HTTP messages. It's implemented as a Java HTTPServlet object. The URL for the HTTP listening connector is http://*gatewayserver*/PSIGW/HttpListeningConnector, where *gatewayserver* is the machine name and port, host name, or IP address of the web server hosting the gateway.

The HTTP listening connector accepts compressed and base 64-encoded data.

PeopleSoft HTTP Message Parameters

You must specify several required parameters in messages that you send to the HTTP listening connector. There are also several optional parameters.

These parameters, also known as *credentials,* are metadata specific to each message that the HTTP listening connector processes. These parameters supply authentication information and descriptive details about how the message is processed. For each message that you send to the connector, PeopleSoft Integration Broker uses the parameters that you supply to create an IBRequest that it uses to process and service the request internally. The following table describes the parameters:

Parameter	***Description***
OperationName	Specify the external alias name.
OperationType	(Optional.) Specify the type of message that is sent. Values are: • *Sync:* The message is synchronous. • *Async:* The message is asynchronous. • *Ping:* The message is used to ascertain whether the target node is active or inactive.
From	Specify the name of the node sending the request. This field is not required if you are invoking SSL encryption and addressing an HTTPS URL.

Parameter	*Description*
Password	Enter the password as it appears in the target node's definition for the source node. The target node authenticates the password when it receives the message. This parameter is required only if password authentication is enabled for the source node definition in the target database.
OrigUser	(Optional.) Specify the user ID from which the message was initially generated.
OrigNode	(Optional.) Specify the name of the node that started the process.
OrigProcess	(Optional.) Specify the name of the process on the source system that sent the message. For example, a message published from the Inventory Definitions page has a process name of INVENTORY DEFIN.
OrigTimeStamp	(Optional.) Specify the time at which the original request was created.
FinalDestination	(Optional.) Specify the name of the node that will ultimately receive the message. This is common when a PeopleSoft Integration Broker hub is used.
To	Specify the name of the node that will receive the message. This parameter is optional if you specified a default target node using the Default Application Server Jolt connect string properties in the integrationGateway.properties file. See Setting General Connection Properties.
SubQueue	(Optional.) Specify the name of a partitioning subqueue to be created at runtime for the message. All messages with the same value for this parameter will be processed in the same subqueue. Unlike the subqueue created by selecting partitioning fields in a queue definition, the subqueue that you specify here has no qualifying criteria except the name that you enter. Field-based partitioning is ignored for messages with this parameter. See “Applying Queue Partitioning” (Integration Broker).

Parameter	*Description*
NonRepudiation	(Optional.) Specify whether the message content in the request should be processed using nonrepudiation logic. Values are: • *Y:* Use nonrepudiation logic. • *N:* Don't use nonrepudiation logic.
MessageName	(Optional.) Specify the name of the message. This parameter is used for backward compatibility with previous PeopleTools releases.
MessageVersion	(Optional.) Specify which version of the message is sent. This parameter is used for backward compatibility with previous PeopleTools releases.
ExternalMessageID	(Optional.) Unique identifier for a message. The ID must not exceed 70 characters. See Using External Message IDs later in this section.

The PeopleSoft HTTP message parameters can be passed with inbound messages to the HTTP listening connector using several methods, and they are transmitted with outbound messages by the HTTP target connector.

See Complying With Message Formatting and Transmission Requirements.

Using External Message IDs

You can specify an external message ID as a parameter in the HTTP listening connector to uniquely identify a message in PeopleSoft Integration Broker, thus ensuring that no duplicate messages are delivered to the system. The ExternalMessageID parameter is optional, but if you do specify this parameter, it must be unique and contain no more than 70 characters.

The HTTP listening connector can receive an external message ID in:

- Query strings.
- HTTP headers.
- SOAPAction headers.
- PeopleSoft IBRequest XML.

The following example shows passing an external message ID in a query string:

```
http://localhost/PSIGW/HttpListeningConnector?From=QE_UNDERDOG&To=
QE_LOCAL&Operation=QE_SYNC_MSG.VERSION_1
```



```
ExternalMessageID=UniqueId0006
```

The following example shows passing an external message ID in an HTTP header:

```
ExternalMessageID: UniqueId0006
```

The following example shows passing an external message ID in a SOAPAction header:

```
http://example.com/QE_SYNC_MSG/QE_UNDERDOG/password/QE_LOCAL/
UniqueId0006
```

The following example shows passing an external message ID in PeopleSoft IBRequest XML:

```
<?xml version="1.0"?>
<IBRequest>
    <From>
        <RequestingNode>QE_UNDERDOG</RequestingNode>
        <OrigTimeStamp>2003-09-29T00:37:30.790-0800</OrigTimeStamp>
        <ExternalMessageID>UniqueId0006</ExternalMessageID>
    /From>
    <ExternalOperationName>QE_SYNC_MSG.VERSION_1</ExternalOperationName>
    <OperationType>sync</OperationType>
    <To>
        <DestinationNode>QE_LOCAL</DestinationNode>
    </To>
    <ContentSections>
        <ContentSection>
            <Headers>
            <version>VERSION_1</version>
            </Headers>
            <Data><![CDATA[<?xml version="1.0"?><QE_SYNC_MSG/>]]></Data>
        </ContentSection>
    </ContentSections>
</IBRequest>
```

Using the HTTP Listening Connector to Receive Message Segments

The HTTP listening connector is segment-aware and you may use it to receive message segments from integration partners.

See “Working With Message Segments” (Integration Broker).

Using the HTTP Target Connector

The HTTP target connector enables you to exchange messages with non-PeopleSoft systems using the HTTP protocol. The HTTP target connector uses SSL for all basic security services, including client-side authentication.

The HTTP target connector also supports the Simple Object Access Protocol (SOAP) XML format.

The connector ID for the HTTP target connector is *HTTPTARGET.*

IBInfo Data Contained in HTTP Headers

A message has two parts—the transaction data and the IBInfo header that is the routing envelope used by PeopleSoft Integration Broker. In the event that a receiving system wants to make use of the IBInfo data, IBInfo header information is included when publishing messages to non-PeopleSoft systems when using the HTTP target connector or the JMS target connector.

When using the HTTP target connector to send messages to non-PeopleSoft systems, the following IBInfo data is contained in the HTTP headers. The content of the message (message body) is not impacted.

- ExternalOperationName
- OperationType
- OrigTimeStamp
- NonRepudiation
- To
- From

Gateway-Level Connector Properties

The HTTP target connector provides the option of routing through proxy servers. To enable this capability, you must set the domain name of the proxy server and the port number of the proxy server in the integrationGateway.properties file:

See Running Integration Gateways Behind Proxy Servers.

Node-Level Connector Properties

The HTTP target connector features properties that correspond to standard HTTP 1.1 header fields, as well as several custom properties that are documented in the following table. The World Wide Web Consortium (W3C) web site provides complete documentation for the standard header fields.

See World Wide Web Consortium

Property ID	***Property Name***	***Description***
HTTPPROPERTY	Method	Specify the HTTP method used to send messages. The valid values are: • *POST* (the default). • *GET.*
HTTPPROPERTY	RemoveSOAP-Response	(Optional.) Remove the SOAP wrapping from response messages returned from a third party. The valid values are: • *Y.* (Default.) The SOAP formatting is removed from response messages. • *N.* The SOAP formatting is not removed from response messages.

Property ID	*Property Name*	*Description*
HTTPPROPERTY	SOAPUpContent	(Optional.) Automatically wrap outbound transactions in SOAP 1.1 format. The valid values are: • *Y*. (Default.) Outbound messages are wrapped in SOAP 1.1 format. • *N*. Outbound message are not SOAP-wrapped. To wrap outbound transactions in SOAP 1.2 format, use the SOAP 1.2 parameter described elsewhere in this table.
HTTPPROPERTY	SOAP 12	(Optional.) Use this property in conjunction with the SOAPUpContent parameter described elsewhere in this table. When using the SOAPUpContent parameter to wrap outbound transactions in SOAP format, by default the system wraps the content in SOAP 1.1 format. To wrap the content in SOAP 1.2 format, use the SOAP 1.2 parameter in addition to the SOAPUpContent parameter. The valid values are: • *Y.* Outbound messages are wrapped in SOAP 1.2 format. • *N.* Outbound messages are wrapped in SOAP 1.1 format.
HTTPPROPERTY	Use-WSA10-Namespace	(Optional.) Override the WS-Addressing namespace URI used in outbound messages that have been SOAP wrapped via the connector property SOAPUpContent. The valid values are: • *Y.* (Default.) The namespace URI http://www.w3.org/2005/08/addressing is used. • *N.* The system default WS-Addressing namespace URI is used.

Property ID	*Property Name*	*Description*
PRIMARYURL	URL	Specify the URL to which messages are sent using this connector.
BACKUPURL	URL	(Optional.) Specify the URL to which messages can be sent if the primary URL is inaccessible.
HEADER	SendUncompressed	Specify whether to send messages decompressed. Options are: • *Y:* Send messages decompressed and decoded. (Default.) • *N:* Send messages compressed and base64 encoded.
HEADER	Proxy-Authorization	Specify the user ID and password for proxy authentication. See Running Integration Gateways Behind Proxy Servers.
HEADER	SOAPAction	(Optional.) Enable third-party systems (for example, Universal Description, Discovery, and Integration (UDDI) sites) to receive SOAP transactions over HTTP. The default value is ”“ (a null string).
HEADER	TimeOut	Specify the time in milliseconds for the connector to wait for the message to transmit. If the timeout period expires without a successful transmission, the transaction fails. The default value is *50000* (50 seconds).

Using the Content-Type Property

One of the optional gateway-level properties you can set for the HTTP target connector is Content-Type.

When the HTTP target connector property Content-Type is *application/x-www-form-urlencoded,* the connector converts the content string to MIME format.

Encoding Strings

When encoding a string, the following rules apply:

- The alphanumeric characters "a" through "z", "A" through "Z" and "0" through "9" remain the same.
- The special characters ".", "-", "*", and "_" remain the same.
- The space character " " is converted into a plus sign "+".
- All other characters are unsafe and are first converted into one or more bytes. Then each byte is represented by the three-character string "%xy," where xy is the two-digit hexadecimal representation of the byte.

Using the HTTP Target Connector to Send Message Segments

The HTTP target connect is segment-aware and you may use it to send message segments to integration partners.

See “Working With Message Segments” (Integration Broker).

Complying With Message Formatting and Transmission Requirements

This section discusses:

- The PeopleSoft XML message wrapper.
- The PeopleSoft non-XML message element.
- Passing HTTP parameters.
- Specifying message destinations in HTTP headers.
- Adding nonrepudiation signatures.
- Submitting cookies in the HTTP header.
- Responses to inbound request messages.
- Submitting SOAP messages.

This section directly addresses the issue of third parties that format and transmit messages to the HTTP listening connector; third parties should also expect the HTTP target connector to format and transmit outbound messages using the same standards.

The PeopleSoft XML Message Wrapper

At a minimum, when you submit message content to the HTTP listening connector, you submit it—preceded by the following XML version declaration—inside a simple XML wrapper:

```
<?xml version="1.0"?><![CDATA[your_message_content]]>
```

Upon receiving the message, the integration gateway strips off the outer elements, leaving the message content with its original XML version declaration to be handled by PeopleSoft Integration Broker:

```
<?xml version="1.0"?>your_message_content
```


The message content can comply with the PeopleSoft rowset-based message format, which you can manipulate using the PeopleCode Rowset class. It can also be nonrowset-based XML-DOM-compliant data, which you can manipulate with nonrowset PeopleCode. Both formats are compatible with Application Engine transform programs, in which you can manipulate the message content using both PeopleCode and Extensible Stylesheet Language Transformation (XSLT) code.

The following template shows how a message in PeopleSoft rowset-based message format fits into the XML wrapper (data omitted for readability):

```
<?xml version="1.0"?><![CDATA[<?xml version="1.0"?>
      <psft_message_name><FieldTypes>...</FieldTypes><MsgData>...</MsgData>
      </psft_message_name>]]>
```

Note: *Psft_message_name* is the name of the message definition in the PeopleSoft database.

The PeopleSoft Non-XML Message Element

If you're submitting a non-XML message, you must insert the message content into a special element containing its own CDATA tag, as follows:

```
<?xml version="1.0"?><![CDATA[<?xml version="1.0"?>
      <any_tag psnonxml="yes">
         <![CDATA[your_nonXML_message_content]]>
      </any_tag>]]>
```

Note: *Any_tag* can be any tag that you want to use. This is an XML-DOM-compliant method of transmitting non-XML data.

The following restrictions apply to the content of non-XML messages, such as those in comma-separated value (CSV) or PDF format:

- If the message content is non-XML text, it must be encoded as characters that are compliant with Unicode Transformation Format 8 (UTF-8).
- If the message content is non-text (binary), it must be encoded in base64 format.

Upon receiving the message, the integration gateway strips off the outer elements, leaving the non-XML message content inside a valid XML-DOM-compliant wrapper with its original XML version declaration.

Passing HTTP Parameters

You can pass parameters to the HTTP listening connector in:

- The PeopleSoft message wrapper, through an HTTP POST.
- The HTTP header, through an HTTP GET or POST.
- The URL query string, through an HTTP GET or POST.

The only HTTP parameters that you must provide for basic messaging are MessageName and RequestingNode. If you pass them in the PeopleSoft message wrapper, you must embed them in an XML structure along with the CDATA element containing the message content. Following is the minimum wrapper structure required to pass the parameters this way:

```
<?xml version="1.0"?>
<IBRequest>
   <ExternalOperationName>psft_operation_name</ExternalOperation
```



```
    Name>
   <From><RequestingNode>psft_node_name</RequestingNode>
   </From>
   <ContentSections>
      <ContentSection>
         <Data><![CDATA[<?xml version="1.0"?>your_message_content]]>
         </Data>
      </ContentSection>
   </ContentSections>
</IBRequest>
```

Note: *Psft_message_name* and *psft_node_name* are the names of the message definition and the sending system's node definition in the PeopleSoft database.

If you want to pass all of the HTTP message parameters in the PeopleSoft message wrapper, you embed them in the XML wrapper structure as follows (required parameters are shown emphasized, and element values are omitted for readability):

```
<?xml version="1.0"?>
<IBRequest>
   <ExternalOperationName/>
   <OperationType/>
   <From><RequestingNode/>
      <Password/>
      <OrigUser/>
      <OrigNode/>
      <OrigProcess/>
      <OrigTimeStamp/>
   </From>
   <To>
      <FinalDestination/>
      <DestinationNode/>
      <SubChannel/>
   </To>
   <ContentSections>
      <ContentSection>
         <NonRepudiation/>
         <MessageVersion/>
         <Data><![CDATA[<?xml version="1.0"?>your_message_content]]>
         </Data>
      </ContentSection>
   </ContentSections>
</IBRequest>
```

The following template shows the format for passing HTTP message parameters in the HTTP message header. The optional parameters can be omitted if not needed. The HTTP header format is as follows (required parameters are shown emphasized):

```
OperationName: OperationName
OperationType: sync|async|ping
From: RequestingNode
Password: Password
OrigUser: OrigUser
OrigNode: OrigNode
OrigProcess: OrigProcess
OrigTimeStamp: OrigTimeStamp
FinalDestination: FinalDestination
To: DestinationNode
SubQueue:  SubQueue
NonRepudiation: Y|N
```

Warning! Whether you send message parameters in the message wrapper or in the HTTP header, those parameters—including the password—aren't secure if you don't encrypt the message. You can secure messages by implementing SSL encryption.

The following template shows the format for passing HTTP message parameters in a URL query string. Include all of the parameter variables, even if you don't supply values for some of them. With only the required parameters, the URL query string looks like the following (required parameters are emphasized):

```
http://gatewayserver/PSIGW/HttpListeningConnector?&Operation=Operation
Name&OperationType=&From=RequestingNode&Password=&OrigUser=&OrigNode=
&OrigProcess=&OrigTimeStamp=&FinalDestination=&To=&SubQueue=
&NonRepudiation=&MessageVersion=
```

The full URL query string format is:

```
http://gatewayserver/PSIGW/HttpListeningConnector?&Operation=Operation
Name&OperationType=[sync|async|ping]&From=RequestingNode&Password=
Password&OrigUser=OrigUser&OrigNode=OrigNode&OrigProcess=OrigProcess&
OrigTimeStamp=OrigTimeStamp&FinalDestination=FinalDestination&To=DestinationNode&Su⇒

bQueue=SubQueue&NonRepudiation=[Y|N]&MessageVersion=MessageVersion
```

Warning! URL query strings are always transmitted in clear text, so your parameters are visible to the world. This means that using a query string to send message parameters—such as a password—is highly insecure. Consequently, it is not recommended.

Using an HTTP POST is the only way that you can send message content to PeopleSoft Integration Broker through the HTTP listening connector. However, you can use an HTTP GET when you don't need to post message content. In this case, you pass the HTTP connector properties in the URL query string or in the HTTP header, but you don't insert any message content or XML wrapper. For example, you might have requests for information (queries), such as a request for a customer list. In this case, you need to specify only the message name (for example, CUSTOMER_LIST_REQUEST) and the name of the requesting node in the URL query string or the HTTP header.

Specifying Message Destinations in HTTP Headers

When message credentials are supplied in HTTP headers, the "To:" (destination node) specification is ignored. PeopleSoft Integration Broker uses the Default Application Server node entry in the integrationGateway.properties file as the destination node, not the "To:" entry from the headers. If no default application server entry is specified in the integrationGateway.properties file, the follow error is generated:

```
<?xml version="1.0"?>
<IBResponse type="error">
<DefaultTitle>Integration Broker Response</DefaultTitle>
<StatusCode>20</StatusCode>
<MessageID>10201</MessageID>
<DefaultMessage>null</DefaultMessage>
</IBResponse>
```

You can specify destination node information in the SOAPAction field or HTTP query string.

Note: If using SOAP, PeopleSoft Integration Broker takes all IBInfo from the SOAPAction field, not from the HTTP header or HTTP query string.

Adding Nonrepudiation Signatures

If you’re working with a nonrepudiated message, its signature must be located at the same level as the message data.-The message doesn’t need to be formatted with the PeopleSoft rowset hierarchy, as long as it's enclosed in valid XML and has the signature section as specified by the W3C. The following template describes a nonrepudiation signature alongside the PeopleSoft rowset-based format message it represents,

within the ContentSection element of the PeopleSoft XML message wrapper (the tags you must add for nonrepudiation are in bold):

```
<ContentSections>
   <ContentSection>
      <NonRepudiation>Y</NonRepudiation>
          <Data><?xml version="1.0"?><any_tag><data>
               <![CDATA[<?xml version="1.0"?>your_message_content]]></data><Signatu⇒

re><SignedInfo><CanonicalizationMethod/><SignatureMethod/><Reference><DigestMethod>⇒

<DigestValue>...</DigestValue></DigestMethod></Reference></SignedInfo><SignatureVal⇒

ue>...</SignatureValue></Signature></any_tag>
      </Data>
   </ContentSection>
</ContentSections>
```

Note: *Any_tag* can be any tag that you want to use, such as *My_NR_Message*.

You can find more information about the proposed standard for XML signature syntax and processing at the W3C web site.

See XML Signature Syntax and Processing

Important! In PeopleSoft Integration Broker, all signatures use line feeds for newlines, so the nonrepudiation signature cannot include any carriage return and line feed (CR/LF) pairs. A non-PeopleSoft application must strip out the carriage returns before inserting the signature and sending the message.

Note: To handle nonrepudiated messages, you must install node-based digital certificates on the sending and receiving systems and configure the message and channel definitions to use the nonrepudiation feature.

See Implementing Nonrepudiation.

Submitting Cookies in HTTP Headers

The HTTP listening connector supports cookies. Cookies that are passed as part of a message request to the HTTP listening connector are processed, read, and manipulated by the receiving PeopleCode in the application. You enter cookies in the HTTP message header. For example:

```
Cookie: favoritecolor=green; path=/; expires Mon, 10-Dec-2007 13:46:00 GMT
```

In this example, the header entry would result in a cookie named favoritecolor. The value of favoritecolor is *green*. This cookie has a path of /, meaning that it is valid for the entire site, and it has an expiration date of December 10, 2007 at 1:46 p.m. Greenwich Mean Time (GMT).

See “Handling Cookies” (Integration Broker).

Responses to Inbound Requests

PeopleSoft Integration Broker responds to inbound requests in one of three ways:

- For a successfully received *synchronous* transmission, the integration gateway passes the request to the integration engine.

The integration engine generates and passes back through the listening connector a response in a format determined by the applicable node, service operation definition and routing definition for the request.

- For a successfully received *asynchronous* transmission, the integration gateway immediately returns a simple XML acknowledgment message.

  The following example shows a successful asynchronous acknowledgment:

```
<?xml version="1.0"?>
<IBResponse type="success">
   <DefaultTitle>Integration Broker Response</DefaultTitle>
   <StatusCode>0</StatusCode>
   <TransactionID>UNDERDOG.QE_SALES_ORDER_ASYNC_CHNL.20</TransactionID>
</IBResponse>
```

- For an *unsuccessful* transmission, the integration gateway immediately returns a simple XML error message in a standard XML error format for all requests (except SOAP requests), if error handling is invoked in the integration gateway.

  The following is an example of this standard error response:

```
<?xml version="1.0"?>
<IBResponse type="error">
   <DefaultTitle>Integration Broker Response</DefaultTitle>
   <StatusCode/>
   <MessageID/>
   <DefaultMessage/>
   <MessageParameters>
      <Parameter/>
   </MessageParameters>
</IBResponse>
```

Submitting SOAP Messages

SOAP messages support a subset of the HTTP message parameters— two required parameters and two optional parameters. You pass them to the HTTP listening connector in a SOAP-specific HTTP header. Concatenate them in a string, with each parameter preceded by a forward slash (/). They must appear in the following order:

```
http://example.com/OperationName/RequestingNode/Password/DestinationNode
```

The following example shows where the parameter string belongs in a SOAP HTTP header:

```
POST /get_BindingDetail HTTP/1.1
Host: www.someOperator.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: http://example.com/PURCHASE_ORDER/MY_NODE/
PSFT_PASS/PSFT_NODE
```

Because the last two parameters are optional, you can exclude them; however, you must still include the forward slashes. This example excludes the password:

```
SOAPAction: http://peoplesoft.com/PURCHASE_ORDER/MY_NODE//PSFT_NODE
```

Consider the following points when submitting SOAP messages:

- The SOAPAction must always be in the HTTP header, not contained within the IBRequest XML.

- The SOAPAction format from previous PeopleTools releases is still supported. The format from previous releases has the parameters concatenated in a string separated by pound signs ("#"): SOAPAction: #PURCHASE_ORDER#MY_NODE#PSFT_PASS#PSFT_NODE
- For SOAP 1.2 requests the value is taken from the "action" in the Content-Type Content-type: application/soap+xml; action=<value used for SOAPAction> since the SOAPAction HTTP header is not used in SOAP 1.2.

Warning! When you send message parameters in the SOAP header, those parameters—including the password—aren't secure if you don't encrypt the message. You can secure messages by implementing SSL encryption.

If an error occurs on the integration gateway during processing, a SOAP-specific XML error is generated instead of a standard XML error. Following is an example of an error in SOAP-specific XML format:

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Server</faultcode>
         <faultstring>Server Error</faultstring>
         <detail>
            <IBResponse type="error">
               <DefaultTitle>Integration Broker Response</DefaultTitle>
               <StatusCode>10</StatusCode>
               <MessageID>10731</MessageID>
               <DefaultMessage></DefaultMessage>
            </IBResponse>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Understanding HTTP Status Codes

This section describes HTTP status codes for non-SOAP and SOAP messages.

Status Codes for Non-SOAP Messages

The following list summarizes HTTP status codes for non-SOAP messages:

- For an asynchronous message, HTTP status codes *200* to *299* indicate a message status of Success.
- For a synchronous message, the HTTP status code *200* indicates a message status of Success.
- HTTP status code *404* indicates that the server has not found anything matching the Request-URI. In this case, an ExternalSystemContactException is generated on the integration gateway and the message status goes to Retry.
- HTTP status code *503* indicates that the server is currently unable to handle the request due to temporary server overload or maintenance. In this case, an ExternalSystemContactException is generated on the integration gateway and the message status goes to Retry.
- All other HTTP status codes generate an ExternalApplicationException. The status of these messages goes to Error.

Status Codes for SOAP Messages

This section summarizes HTTP status codes for SOAP messages.

If you are following SOAP 1.1 standards, the HTTP status code 500 indicates an Error.

If you are following SOAP 1.2 standards, the following HTTP status codes apply:

- HTTP status code 400 can mean any of the following:
  - InvalidMessageException
  - MessageMarshallingException
  - MessageUnmarshallingException
  - Malformed HTTP/XML
- HTTP status code 405 indicates that the method is not POST.
- HTTP status code 415 indicates the content type is not text/xml.
- HTTP status code 500 can mean any of the following:
  - ExternalSystemContactException
  - ExternalApplicationException
  - GeneralFrameworkException

Running Integration Gateways Behind Proxy Servers

When a proxy server is set up for a network on which the integration gateway resides, all HTTP transactions are routed through that proxy server automatically. The HTTP transport layer uses proxy server settings that you specify in the integrationGateway.properties file. The message is routed to the proxy server and then on to the internet. Only the HTTP target connector can use a proxy server.

Inserted in the HTTP message header of each transaction is a *Proxy-Authorization* header field containing a user ID and password. The proxy server uses these values to authenticate the message and then passes it on to its target.

PeopleSoft also enables you to exclude user-defined hosts from connecting through a proxy server.

Setting Proxy Web Server Properties

To run the integration gateway behind a proxy server:

1. Set the gateway-level properties.

   Uncomment and add values for the properties in the integrationGateway.properties file section labeled *Proxy webserver section*.

Property	*Description*
ig.proxyHost	Enter the domain name of the proxy web server; for example: `proxy.example.com`
ig.proxyPort	Enter the port number of the proxy web server; for example: `80`

ig.nonProxyHost	Enter a list of hosts that should be accessed directly, instead of through the proxy server. The values can be a list of hosts, each separated by a |, and in addition a wildcard character (*) can be used for matching. For example: For example:
	`ig.nonProxyHosts=*.google.com|finan⇒`
	`ce.yahoo.com`

The HTTP target connector reads these two properties and calls the setProxy function. In an outbound transaction, the request is redirected to the proxy server and the proxy server forwards the request to the destination URL.

2. Set the node-level property.

   You set the user ID and password required by the proxy server in the *HEADER, Proxy-Authorization* connector property. The integration gateway encodes the values that you provide, adds the required formatting, and sends it. The format is:

   userid:*password*

Related Links

Using the integrationGateway.properties File

Working With the JMS Connectors

This section provides an overview of the JMS connectors and discusses how to:

- Specify JNDIFactory class names.
- Use the JMS listening connector.
- Use the JMS target connector.

Understanding the JMS Connectors

The JMS listening and target connectors enable communication between JMS provider systems and an integration gateway using standard JMS protocols. PeopleSoft currently supports Java Native Directory Interface (JNDI) only for File System Context [fscontext] and RMI lookup.

Note: Check My Oracle Support for the JMS specification currently supported by PeopleTools. PeopleSoft Integration Broker's JMS listening connector and JMS listening connector are compliant with the specification version listed.

Supported JMS Providers

To use the JMS connectors, you must add specific Java archive (JAR) files to the Java CLASSPATH. The JAR files that you add to the CLASSPATH depend on the JMS provider with which you're communicating. The following JMS providers are supported:

JMS Provider	*Required Files*
Oracle WebLogic	N/A
IBM MQ Series	jms.jar, jndi.jar, fscontext.jar, com.ibm.mqjms.jar

Note: Not only can a gateway running on a Oracle WebLogic web server communicate with a WebLogic JMS provider, but both services can run on a single installation of WebLogic. However, the gateway still treats the JMS provider as a separate system, and it must be configured the same way as in any other scenario.

You can also add generic JMS providers for use with PeopleSoft Integration Broker.

See Adding Generic JMS Providers.

Integrations with Oracle SOA B2B Suite

The JMS target connector and JMS listening connector feature properties that enable you to integrate with Oracle SOA B2B Suite.

The Oracle SOA-B2B server supports several industry-standard e-commerce protocols as well as several transports for message delivery.

See Oracle B2B Documentation

PeopleSoft's integration to the Oracle SOA B2B Suite uses the JMS transport to deliver and receive messages, and as such the JMS target connector and JMS listening connector are used.

For outbound integrations with Oracle SOA B2B you set JMS target connector properties at the node-level. For inbound integrations with Oracle SOA B2B you set JMS listening connector properties in the integration gateway properties file. Setting these properties is discussed elsewhere in this section.

See Using the JMS Target Connector, Using the JMS Listening Connector

Specifying JNDIFactory Class Names

You must set up the JNDIFactory class names for the JMS provider in the section of the integrationGateway.properties file labeled *JMS configuration Section.*

When you set the JMSProvider property, the provider name that you enter must match the provider in the JNDIFactory class name exactly. You must set this property for both the JMS listening connector and the JMS target connector. This property is case-sensitive.

JMS Provider	***Property***	***Description***
Oracle WebLogic	ig.jms.JMSProvider.JNDIFactory.Weblogic	Specify the JNDIFactory class name for a Oracle WebLogic JMS provider. The default value is: *weblogic.jndi.WLInitialContextFactory*
IBM MQ Series	ig.jms.JMSProvider.JNDIFactory.MQSeries	Specify the JNDIFactory class for an MQSeries JMS provider. The default value is: *com.sun.jndi.fscontext.RefFSContextFactory*

You can also specify a service provider that is not listed. For example, if you are using MSMQ, enter the following value for the property:

```
ig.jms.JMSProvider.JNDIFactory.MSMQ=com.sun.jndi.fscontext.RefFSContextFactory
```

Using the JMS Listening Connector

The JMS listening connector has two components: a subscriber and a queue listener. The JMS subscriber subscribes to different topics and the JMS queue listens on queues for new messages.

Note: The JMS listening connector always expects JMS messages in text format.

Receiving Messages

The JMS listening connector retrieves topics and queues that you have defined in integrationGateway.properties file. For each topic it starts a topic subscriber, and for each queue it starts a queue listener. When a message arrives either for a queue or topic, the JMS listening connector sends the message to the integration engine.

A parameter called ExternalMessageID is used to ensure that messages are received only once. When the JMS listening connector receives a message, it sets an external message ID in IBInfo and sends this information to the PeopleSoft Integration Broker with the message content. If the external message ID exists in IBInfo, the application server checks for duplicate messages. If a duplicate is found, an error is generated. The external message ID is optional. If specified, it must be unique and not exceed 70 characters.

Securing Messages to JMS Queues

PeopleSoft Integration Broker does not perform security validation checks on messages transmitted to JMS queues.

Note: JMS administrators must set up secure queues on providing systems.

Error Handling

If an error occurs during message processing, the JMS listening connector publishes the message back to either an error topic or an error queue. All error messages feature a header called ErrorDescription which contains a description of the error.

Note: If the application server returns the status *20,* the message is published to the error topic and the response is logged in the integration gateway message log.

To capture errors you must set error topic or error queue properties in the JMS Configuration Section of the integrationGateway.properties file. These properties are discussed later in this section. If both an error topic and an error queue are set up and configured, only the error queue will capture error messages.

JMS Queue Listener Properties

You can configure multiple queues in the section of the integrationGateway.properties file labeled *JMS Configuration Section.* To configure multiple queues, use the convention, *ig.queue1*, *ig.queue2*, *ig.queue3*, and so on.

Property	***Description***
ig.jms.Queues	Specify the number of queue listeners to instantiate.
ig.jms.Queue1	Specify the queue name.
ig.jms.Queue1.Provider	Specify the queue provider name.
ig.jms.Queue1.JMSFactory	Specify the JMSFactory name that is bound to JNDI for the queue.
ig.jms.Queue1.MessageSelector	(Optional.) Specify the message filter.
ig.jms.Queue1.URL	Specify the JMS provider's URL to JNDI.
ig.jms.Queue1.User	(Optional.) Specify the JMS queue user name.
ig.jms.Queue1.Password	(Optional.) Specify the JMS queue password. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
ig.jms.Queue1.MessageName	This property is desupported and is being maintained for backwards compatibility only.

Property	*Description*
ig.jms.Queue1.MessageVersion	This property is desupported and is being maintained for backwards compatibility only.
ig.jms.Queue1.OperationName	(Optional.) Specify the name of the service operation and the service operation version. The format is: *ig.jms.Queue1.OperationName=OperationName.OperationVersion.*
ig.jms.Queue1.RequestingNode	(Optional.) Specify the name of the requesting node.
ig.jms.Queue1.DestinationNode	(Optional.) Specify the name of the destination node.
ig.jms.Queue1.NodePassword	(Optional.) Specify the password for the requesting node. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
ig.jms.Queue1.SubChannel	(Optional.) Specify the name of the subchannel. Messages published to this queue go to the subchannel indicated.

JMS Topic Subscriber Properties

You can configure multiple topics, in the section of the integrationGateway.properties file labeled *JMS configuration Section.* To configure multiple topics, use the convention *ig.topic1*, *ig.topic2*, *ig.topic3*, and so on.

Property	*Description*
ig.jms.Topics	Specify the number of topic subscribers to instantiate.
ig. jms.Topic1	Specify the topic name.
ig. jms.Topic1.Provider	Specify the topic provider name.
ig. jms.Topic1.JMSFactory	Specify the JMSFactory name that is bound to JNDI for the topic.
ig. jms.Topic1.MessageSelector	(Optional.) Specify the message filter.
ig. jms.Topic1.URL	Specify the JMS provider's URL to JNDI.

Property	*Description*
ig. jms.Topic1.User	(Optional.) Specify the JMS topic user name.
ig. jms.Topic1.Password	(Optional.) Specify the JMS topic password. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
ig.jms.Topic1.MessageName	This property is desupported and is being maintained for backwards compatibility only.
ig.jms.Topic1.MessageVersion	This property is desupported and is being maintained for backwards compatibility only.
ig.jms.Topic1.OperationName	(Optional.) Specify the name of the service operation and the service operation version. The format is: *ig.jms.Queue1.OperationName=OperationName.OperationVersion.*
ig.jms.Topic1.RequestingNode	(Optional.) Specify the name of the requesting node.
ig.jms.Topic1.DestinationNode	(Optional.) Specify the name of the destination node.
ig.jms.Topic1.NodePassword	(Optional.) Specify the password for the requesting node. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
ig.jms.Topic1.SubChannel	(Optional.) Specify the name of the subchannel. Messages published to this topic go to the subchannel indicated.

Error Queue Properties

To capture JMS listening connector errors in an error queue, set the following properties in the *JMS Configuration Section* of the integrationGateway.properties file.

Property	*Description*
ig.jms.ErrorQueue	Specify the name of queue to which error messages are published.
ig.jms.ErrorQueue-Provider	Specify the name of the JMS provider.

Property	*Description*
ig.jms.ErrorQueue-User	(Optional.) Specify the JMS error queue user name.
ig.jms.ErrorQueue-Password	(Optional.) Specify the JMS error queue password. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
ig.jms.ErrorQueue-JMSFactory	Specify the queue connection factory name.
ig.jms.ErrorQueue-Url	Specify the JMS provider's URL to JNDI.

Error Topic Properties

To capture JMS listening connector errors in an error topic, set the following properties in the *JMS Configuration Section* of the integrationGateway.properties file.

Property	*Description*
ig.jms.ErrorTopic	Specify the name of topic to which error messages are published.
ig.jms.ErrorTopic-Provider	Specify the name of the JMS provider.
ig.jms.ErrorTopic-User	(Optional.) Specify the JMS error topic user name.
ig.jms.ErrorTopic-Password	(Optional.) Specify the JMS error topic password. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
ig.jms.ErrorTopic-JMSFactory	Specify the JNDIFactory name.
ig.jms.ErrorTopic-Url	Specify the JMS provider's URL to JNDI.

JMS Listening Connector Properties for Integrating with Oracle SOA B2B Suite

For inbound integrations from Oracle SOA B2B Suite, you must set the following property in the integrationGateway.properties file:

```
ig.AS2.<FROM_PARTY>.<TO_PARTY>.MessageName=
```


Set the values as follows:

- **FROM_PARTY.** Enter the sending node name.
- **TO_PARTY.** Enter the receiving node name.
- **MessageName.** Set this property equal to the message.version of the message.

For example:

```
ig.AS2.SOA_B2B.QE_LOCAL.MessageName=USER_PROFILE.VERSION1
```

In the example, *SOA_B2B* is the sending node, *QE_LOCAL* is the receiving node, and *USER_PROFILE.VERSION1* is the message.version.

Information about setting properties for the JMS target connector for outbound integrations to Oracle SOA B2B is provided elsewhere in this section.

JMS Message Header Properties

For the JMS listening connector to process messages, you must set the following properties. You can set these properties in JMS message headers, the integrationGateway.properties file or in the body of the XML message.

You can specify JMS headers in the integrationGateway.properties file for both queues and topics. However you must be using separate queues or topics per requesting node/message combination.

You must supply the properties listed in the following table in the JMS message header when you publish messages from a JMS provider system to the integration gateway.

Property	***Description***
MessageName	Specify the name of service operation.
RequestingNode	Specify the requesting node name.
FinalDestinationNode	Specify the final destination nodes. If there are no values, set this property to *Null*.
DestinationNode	Specify the destination node names, separated with commas. If there are no values, set to "" (empty string).
NodePassword	Enter the node password. This password must be encrypted. See Encrypting Passwords.

Property	*Description*
SubChannel	(Optional.) Specify the name of a partitioning subqueue to be created for the service operation at runtime. All service operations with the same value for this parameter are processed in the same subqueue. Unlike the subqueues created by selecting partitioning fields, the subqueue that you specify here has no qualifying criteria except the name that you enter. Field-based partitioning is not used for service operations with this parameter. See “Applying Queue Partitioning” (Integration Broker).

The following example shows specifying JMS header properties in the body of an XML message.

```
<?xml version="1.0" ?>
   <IBRequest>
      <ExternalOperationName>JMS_MessageName</ExternalOperationName>
      <OperationType>Async_or_Synch</OperationType>
         <From>
            <RequestingNode>JMS_RequestingNode</RequestingNode>
            <Password>JMS_NodePassword</Password>
            <OrigUser></OrigUser>
            <OrigNode></OrigNode>
            <OrigProcess></OrigProcess>
            <OrigTimeStamp></OrigTimeStamp >
         </From>
         <To>
            <FinalDestination>JMS_FinalDestination</FinalDestination>
            <DestinationNode>JMS_DestinationNode</DestinationNode>
         </To>
         <ContentSections>
            <ContentSection>
               <NonRepudiation></NonRepudiation>
                <Data></Data>
            </ContentSection>
        </ContentSections>
   </IBRequest>
```

When the message received specifies synchronous mode, a reference to the temporary queue or topic must be set in the JMS message header for the JMS listening connector to determination the destination of the message response. The JMS listening connector also sets the JMS correlation ID when it sends the response so the requestor can properly associate the response with its corresponding request.

If any of the message header properties are missing, an error is logged and an error is published to an error topic or error queue. The message that the connector publishes to the error topic has a property call error and is set to *True.* The error message that is published contains the following information: default message, message ID, message set, message parameters, and body of the message sent.

Starting and Shutting Down the JMS Listening Connector

Use the JMS Listening Connector Administration page to start or shut down the JMS listening connector.

See Administering JMS Listening Connectors

Performing Administrative Tasks on JMS Listening Connectors

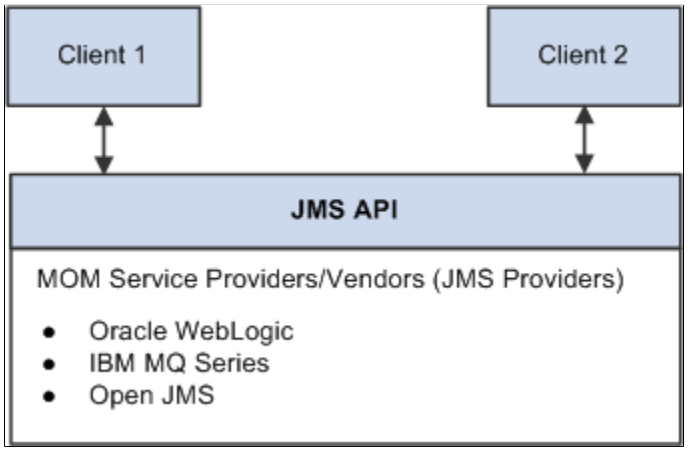
The JMS Listening Connector Administration page enables you to perform administrative tasks on JMS listener queues and subscriber topics, such as start, stop, and pause queues and topics. Using the JMS Listening Connector Administration page to administer JMS listening connectors is discussed elsewhere in the product documentation.

See Administering JMS Listening Connectors

Using the JMS Target Connector

JMS is an application programming interface (API) for accessing message systems. JMS provides a standard Java-based interface to the message services of message-oriented middleware (MOM) providers. The JMS target connector is an adapter to JMS providers, and it can be used with MOM and JMS providers, such as Oracle Weblogic, IBM MQSeries and others. The following diagram illustrates how messages flow through the JMS API:

This diagram illustrations the flow of a message through the JMS API.



The primary features of JMS are.

- Connection factories that are used to create connections to a specific JMS provider.
- Separate publish, subscribe, and point-to-point messaging domains.

  These are defined by separate interfaces so that a provider does not have to support both.
- Topics for publish and subscribe, as well as queues for point-to-point messaging.

When multiple applications must receive the same message, publish and subscribe messaging is used. In publish and subscribe messaging, all of the subscribers subscribe to a topic and all of the publishers publish messages to a topic. The messaging system distributes messages from the publisher to the subscriber. This domain is mainly used for asynchronous messaging.

When one application must send a message to another application, point-to-point messaging is used. This domain is only for synchronous messaging. There are two basic types of point-to-point messaging systems. One uses a client that directly sends a message to another client. The other, more commonly used implementation uses a message queue.

The JMS target connector either publishes a message to a topic or inserts a message into a queue, based on the node-level properties that you set.

The JMS target connector supports only JNDI file context for the lookup of connection factories, topics, and queue names. (Lightweight Directory Access Protocol (LDAP) is not supported.)

The connector ID for the JMS target connector is *JMSTARGET*.

Asynchronous and Synchronous Communication

The JMS target connector provides both synchronous and asynchronous modes of communication. When the node level property ReplyTo is set to *False*, communication is asynchronous. When it is set to *True*, communication is synchronous.

For asynchronous communication, the JMS target connector publishes messages to MOM or drops messages into a queue and commits the session. It does not wait for a response from the destination system. For synchronous communication, after the connector publishes messages or drops them into a queue, it waits for the temporary topic or queue to respond.

For synchronous communication, the exchanges involve only the publisher and a single subscriber. When a JMS-compliant remote node receives a synchronous request message from PeopleSoft, it must use the value of the JMSCorrelationID of the request message to populate the correlation ID of its response message. When the response is received by the PeopleSoft JMS target connector, it compares the JMSCorrelationID of the response message with the JMSCorrelationID of the request. The message is not accepted if these two IDs do not match.

When sending messages either synchronously or asynchronously, the connector sets different string properties in the JMS message header. The properties are used as metadata about the message. The JMS target connector also sets a reference to the temporary queue or topic from which it requires the response.

JMS Target Connector and Message Segments

The JMS target connect is segment-aware and you may use it to send message segments to integration partners.

See “Working With Message Segments” (Integration Broker).

IBInfo Data Contained in JMS Headers

A message has two parts—the transaction data and the IBInfo header that is the routing envelope used by PeopleSoft Integration Broker. In the event that a receiving system wants to make use of the IBInfo data, IBInfo header information is included when publishing messages to non-PeopleSoft systems when using the JMS target connector or the HTTP target connector.

When using the JMS target connector to send messages to non-PeopleSoft systems, the following IBInfo data is contained in the JMS headers. The content of the message (message body) is not impacted.

- RequestingNode
- FinalDestinationNode
- DestinationNodes
- MessageName

- MessageType
- OrigTimeStamp
- NonRepudiation

Gateway-Level Connector Properties

There are no gateway-level JMS target connector properties that you must set.

Node-Level Connector Properties

You must set either a JMS queue or JMS topic for a given node definition. If both are set or are missing the PeopleSoft Integration Broker generates an invalid message exception.

Note: You must register JMS-administered objects—such as topics, queues, and connection factories —that you include as connector properties. The documentation for specific providers should provide instructions on how to register the topics.

JMS message types can be *Text, Map Message, Stream,* or *Object*. However, PeopleSoft provides only text messages. If you need to use other message types, you can write a class that implements the com.peoplesoft.pt.integrationgateway.common.jms.ProcessJMSMessage interface, and you set the class name as a value for JMSMessageTypeClass.

The provider name that you specify for the JMSProvider in the node definition must match the JMSProvider.JNDIFactory property that you specify in the integrationGateway.properties file.

The following table describes the node-level connector properties:

Property ID	*Property Name*	*Description*
JMSTARGET	AS2MODE	This property is used for integrations with Oracle SOA B2B Suite. Set the property value to *Y* to enable integrations with Oracle SOA B2B. Values are: • *Y.* • *N.* (Default.) There are additional JMS target connector properties you must set for integrations with Oracle SOA B2B. See, "JMS Target Connector Properties for Integrations with Oracle SOA B2B" in this section. **Note:** PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification. **Note:** AS2 Connectors will be desupported in future PeopleTools release.
JMSTARGET	JMSAcknowledgement	Specify the acknowledgment type. Values are: • *Auto_Acknowledge*. (Default.) • *Client_Acknowledge.*
JMSTARGET	JMSDeliveryMode	Specify either durable or nondurable delivery. Values are: • *Persistent..* • *Non-persistent.* (Default.)
JMSTARGET	JMSFactory	Specify the factory name. The default value is *QueueConnectionFactory.*
JMSTARGET	JMSMessageTimeToLive	Specify the time in seconds.

Property ID	*Property Name*	*Description*
JMSTARGET	JMSMessageType	Specify the type of message to send. Values are: • *Text* (default). • *MapMessage.* • *Stream.* • *Object.*
JMSTARGET	JMSMessageTypeClass	(Optional.) Specify the implementation class of ProcessJMSMessage. You must set this property when the JMSMessageType is anything other than *Text.*
JMSTARGET	JMSPassword	(Optional.) Specify the password to access the connection. If you choose to specify a password, you must encrypt it. See Encrypting Passwords.
JMSTARGET	JMSPriority	Specify the message priority for delivery. Values range from *0* to *9.* A value of *9* indicates the highest priority. The default is *0.*
JMSTARGET	JMSProvider	Specify the JMS provider's name. Values are: • *MQSeries.* (Default.) • *WebLogic..*
JMSTARGET	JMSQueue	(Optional.) Specify the queue name, if you use a queue. You must use and specify either a topic or a queue.

Property ID	*Property Name*	*Description*
JMSTARGET	JMSReplyTo	Set this property to *True* to receive a response from the external system. Values are: • *True.* • *False* (Default.)
JMSTARGET	JMSTopic	(Optional.) Specify the topic name, if you use a topic. You must use either a topic or a queue.
JMSTARGET	JMSUrl	Specify the URL.
JMSTARGET	JMSUserName	(Optional.) Specify the username to establish a connection to the JMS.
JMSTARGET	JMSWaitForResponse	Specify the time in milliseconds for the connector to wait for the temporary response queue to return a synchronous response message. If a response fails to appear in the queue within the specified period, the transaction fails and the queue is deleted. The default value is *60000* (60 seconds).
JMSTARGET	RemoveSOAP-Response	(Optional.) Remove the SOAP wrapping from response messages returned from a third party. The valid values are: • *Y.* (Default.) The SOAP formatting is removed from response messages. • *N.* The SOAP formatting is not removed from response messages.

Property ID	*Property Name*	*Description*
JMSTARGET	SOAPUpContent	(Optional.) Automatically wrap outbound transactions in SOAP format. The valid values are: • *Y.* (Default.) Outbound messages are wrapped in SOAP format. • N. Outbound messages are not wrapped in SOAP format.
JMSTARGET	Use-WSA10-Namespace	(Optional.) Override the WS-Addressing namespace URI used in outbound messages that have been SOAP wrapped via the connector property SOAPUpContent. The valid values are: • *Y.* (Default.) The namespace URI http://www.w3.org/2005/08/addressing is used. • *N.* The system default WS-Addressing namespace URI is used.
HEADER	SendUncompressed	Specify whether to send messages decompressed. Values are: • *Y:* Send the message decompressed and unencoded. This is the default value. • *N:* Send the message compressed and base 64 encoded.
HEADER	SOAPAction	(Optional.) Enable third-party systems (for example, Universal Description, Discovery, and Integration (UDDI) sites) to receive SOAP transactions over HTTP. The default value is ”“ (a null string).

JMS Target Connector Properties for Integrations with Oracle SOA B2B

The following table lists node-level properties that you must set for the JMS target connector for outbound integrations with Oracle SOA B2B.

Set these properties on the Nodes – Connector page. To access the page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** and click the Connectors tab.

For all properties except for AS2MODE, you must add a new row to the properties grid and manually enter property ID, property name, and property values.

Property ID	***Property Name***	***Description***
JMSTARGET	AS2MODE	This property is used for integrations with Oracle SOA B2B Suite. Set the property value to *Y* to enable integrations with Oracle SOA B2B. Values are: • *Y.* • *N.* (Default.)
JMSTARGET	DOCTYPE_NAME	This property is used for integrations with Oracle SOA B2B Suite. Enter the doctype name. This property is similar to a message name. For example, *purchase_order.*
JMSTARGET	DOCTYPE_REVISION	This property is used for integrations with Oracle SOA B2B Suite. Enter the version of the doctype. For example, *1.0.*
JMSTARGET	FROM_PARTY	This property is used for integrations with Oracle SOA B2B Suite. Enter the name of the sending node.
JMSTARGET	TO_PARTY	This property is used for integrations with Oracle SOA B2B Suite. Enter the name of the node that Oracle SOA B2B uses to route the service operation to the AS2 partner.

Information about setting properties for the JMS listening connector for inbound integrations from Oracle SOA B2B is provided earlier in this topic.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Note: AS2 Connectors will be desupported in future PeopleTools release.

Additional Setup Steps

Before using the JMS target connector, verify that:

1. The JMS messaging system is running.
2. All JMS connection factories, topics, and queues are registered for JNDI lookup.
3. A username and a password are created in the JMS system for use as values for the properties JMSUserName and JMSPassword.

JMS Target Connector Errors and Exceptions

The JMS target connector may generate the following exceptions:

Exception	***Cause***
InvalidMessageException	This exception is generated when any node level or connector parameters are not set properly. Examples are: • Both queue and topic are specified. • Neither queue nor topic is specified. • A JMS security exception is generated. (Verify that the username and password are correct.) • A naming exception occurs.
ExternalApplicationException	This exception is generated when: • The correlation ID does not match when the ReplyTo property is set to *True*. • The message could not put into a queue, or a topic could not be published.
GeneralFrameWorkException	This exception is generated when a naming exception occurs.

Adding Generic JMS Providers

The JMS providers that PeopleSoft supports are Oracle WebLogic, and IBM MQSeries. However, to meet your business requirements you can add generic JMS providers.

This section provides lists of configuration tasks to perform on the JMS listening connector and JMS target connector to add a generic JMS provider to PeopleSoft Integration Broker.

Configuring the JMS Listening Connector for Generic JMS Providers

To configure the JMS listening connector for a generic JMS provider:

- Obtain the following information from the provider:
  - JMS jar file.
  - JNDIFactory information
- Determine if messaging will be in topics or queues.
- Determine if error handling will be in topics or queues.
- Update JMS properties in the integrationGateway.properties file:
  - Update the JNDIFactory entry.

    For example if the provider were Tibco the entry might be:

    ```
    ig.jms.JMSProvider.JNDIFactory.Tibco=com.tibco.JMSFactory
    ```
  - Populate the appropriate messaging topic and queue entries based on how messaging will be handled.
  - Populate the appropriate error topic and queue entries based on how messaging will be handled.

In addition to the information provided in this section, review the JMS Headers Properties section of this topic which discusses the required information that must be in the headers of each message processed by the JMS listening connector.

Configuring the JMS Target Connector for Generic JMS Providers

To configure the JMS target connector for a generic JMS provider:

- Define a node for the provider.
- Assign the JMS target connector to the provider node and specify the target connector properties.

Working With the PeopleSoft 8.1 Connectors

This section provides an overview of the PeopleSoft 8.1 connectors and discusses how to:

- Use the PeopleSoft 8.1 listening connector.
- Use the PeopleSoft 8.1 target connector.

Understanding the PeopleSoft 8.1 Connectors

The PeopleSoft 8.1 listening and target connectors enable communication between PeopleSoft 8.1x applications and an integration gateway using PeopleSoft Application Messaging technology. To the PeopleSoft 8.1x application, the gateway appears to be another PeopleSoft 8.1x application, so no

change in the messaging development process is needed. The connectors also support secure HTTPS communications if SSL encryption is configured on the gateway machine.

Note: The PeopleSoft 8.1 connectors are intended for use for integrations with PeopleSoft systems running PeopleTools 8.1x.

Related Links

"Implementing Web Server SSL/TLS Encryption" (Security Administration)

Using the PeopleSoft 8.1 Listening Connector

In PeopleSoft 8.1x systems, PeopleSoft Application Messaging generates highly structured XML messages that are designed to be sent to PeopleSoft 8.1x Application Messaging gateways. The PeopleSoft 8.1 listening connector mimics the role of the Application Messaging gateway by transparently receiving and processing PeopleSoft 8.1x messages. This connector transforms inbound PeopleSoft 8.1x messages into PeopleSoft Integration Broker formatted XML messages that can be processed by the integration gateway and ultimately by the integration engine. This conversion is necessary because the two message formats are distinctly different.

The URL for the PeopleSoft 8.1 listening connector is http://*gatewayserver*/PSIGW/PS81ListeningConnector, where *gatewayserver* is the machine name and port, host name, or IP address of the web server hosting the gateway.

This connector automatically handles base64–encoded and compressed messages, as well as uncompressed messages.

Using the PeopleSoft 8.1 Target Connector

This connector enables the gateway to communicate with PeopleSoft 8.1x applications that use PeopleSoft Application Messaging technology. It converts outbound messages to the Application Messaging native format. Messages from the PeopleSoft Integration Broker system reach the PeopleSoft 8.1x system through the Application Messaging gateway on the PeopleSoft 8.1x system.

The PeopleSoft 8.1 target connector uses the HTTP target connector to manage the HTTP communication with the PeopleSoft 8.1x Application Messaging gateway. The PeopleSoft 8.1 target connector focuses on messaging semantics, instead of communication details; it constructs an Application Messaging XML document and sends it using the HTTP target connector. The PeopleSoft 8.1 target connector detects the status of returned responses by the value in the ReturnCode field in the XML response.

The connector ID for the PeopleSoft 8.1 target connector is *PSFT81TARGET*.

Gateway-Level Connector Properties

The PeopleSoft 8.1 target connector has one gateway-level property, in the section of the integrationGateway.properties file labeled *DELIVERED CONNECTOR CONFIGURATION Section.* This property specifies where the connector can send messages if a target URL isn't specified in the connector's node-level properties. Specify the URL as follows:

```
ig.connector.amtargetconnector.url=peoplesoft_8.1x_application_messaging_gateway
```

You can override this value by specifying a different URL in the node-level connector properties, in the node definition for the PeopleSoft 8.1x target node, or in the transaction definition for the message.

Node-Level Connector Properties

The following table describes the node-level connector properties:

Property ID	*Property Name*	*Description*
PSFT81TARGET	URL	Specify the PeopleSoft 8.1x Application Messaging gateway URL to which messages are sent using this connector.
HEADER	TimeOut	Specify the time in milliseconds for the connector to wait for the message to transmit. If the timeout period expires without a successful transmission, the transaction fails. The default value is *50000* (50 seconds).

Working With the PeopleSoft Connectors

This section provides an overview of the PeopleSoft connectors and discusses how to:

- Use the PeopleSoft listening connector.
- Use the PeopleSoft target connector.

Understanding the PeopleSoft Connectors

The PeopleSoft listening and target connectors establish the primary connection between a PeopleSoft application's integration engine and its designated local gateway. They also support secure HTTPS communications if SSL encryption is configured on the gateway machine.

Using the PeopleSoft Listening Connector

The PeopleSoft listening connector receives requests from integration participants in the PeopleSoft internal messaging format. Like the HTTP listening connector, the PeopleSoft listening connector is implemented as a Java HTTPServlet object. However, it receives requests in PeopleSoft Multipurpose Internet Mail Extensions (MIME) format. A PeopleSoft integration engine sends messages formatted in MIME over HTTP. The PeopleSoft listening connector receives these messages as POSTs (GET requests cannot be made in this way) and immediately converts the MIME input into a Java string object.

The PeopleSoft listening connector logs these requests and then invokes the gateway manager to unmarshall the string into an IBRequest object. The gateway manager invokes the target connector specified in the ConnectorClassName field in the IBRequest, which is derived from the node definition on the source integration engine. The gateway manager returns the responses to the connector, where they are logged and sent back to the original requesting systems, typically integration engines.

The URL for the PeopleSoft listening connector is http://*gatewayserver*/PSIGW/PeopleSoftListeningConnector, where *gatewayserver* is the machine name and port, host name, or IP address of the web server hosting the gateway.

Note: Third-party applications and PeopleSoft releases that don't include PeopleSoft Integration Broker should not send messages to this connector unless they can produce a properly MIME-encoded, PeopleSoft formatted message.

Using the PeopleSoft Target Connector

The PeopleSoft target connector initiates conversation with a PeopleSoft application's integration engine over a Oracle Jolt connection in the PeopleSoft internal messaging format. The integration gateway sends messages to a specific integration engine based on the destination node specified in an incoming message. Use this connector to send messages only to PeopleSoft applications that use PeopleSoft Integration Broker.

The connector ID for the PeopleSoft target connector is *PSFTTARGET*.

Gateway-Level Connector Properties

There are no gateway-level connector properties specific to this connector; however, it uses both the node-specific and default Oracle Jolt connect string properties in the integrationGateway.properties file to determine where to send the messages.

See Setting General Connection Properties.

Node-Level Connector Properties

There are no node-level connector properties for the PeopleSoft target connector.

Working With the PeopleSoft Services Listening Connector

This section discusses how to:

- Set parameters for the PeopleSoft services listening connector.
- Pass parameters for the PeopleSoft services listening connector.
- Pass parameters to get XML schema, WSDL, and WSIL.

Understanding the PeopleSoft Services Listening Connector

The PeopleSoft services listening connector is used for inbound integrations with web services.

SOAP Messages

If the inbound request is a SOAP message:

- The SOAPAction must take the following format for SOAP 1.1 requests:

```
SOAPAction:<External_alias_name>
```


- For SOAP 1.2 requests, the value of the "action" in the content type should be used:

  ```
  Content-type: application/soap+xml; action=<External_alias_name>
  ```

- The response message should also be in SOAP format. If it is not, it should be wrapped in SOAP format.
- Any errors generated are in SOAP format or wrapped in the SOAP fault tag and returned to the sender.

Setting Parameters for the PeopleSoft Services Listening Connector

The same required and optional parameters that you can set for the HTTP listening connector pertain to the PeopleSoft services listening connector. For a list of the required and optional parameters, see the Using the HTTP Listening Connector section presented previously in this topic.

See Using the HTTP Listening Connector.

Passing Parameters to the PeopleSoft Services Listening Connector

This section discusses how to pass parameters to the PeopleSoft services listening connector.

Passing Parameters to the PeopleSoft Services Listening Connector in URL Query Format

You can pass parameters to the PeopleSoft service listening connector using a URL query string using the following format:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListening
Connector?Operation=OperationName
```

The following format is also supported:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListening
Connector?Operation=<OperationName>>&To=<ReceiverNode>&From=
<SenderNode>&OperationType=<Type>
```

Passing Parameters to the PeopleSoft Services Listening Connector in Path Format

You can pass parameters to the PeopleSoft service listening connector using a path format using the following format:

```
http://192.0.2.10/PSIGW/PeopleSoftServiceListeningConnector/
SERVICE_OPERATION.VERSION.xsd
```

Passing Parameters to Get XML Schema, WSDL and WSIL

You can use query format or path format to get XML schema, WSDL and WSIL.

Using Query Format to Get XML Schema, WSDL and WSIL

Use the following query format to get XML schema:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListeningConnector?Operation=
```



```
GetSchema&xsd=SERVICE_OPERATION.VERSION
```

Use the following query format to get WSDL:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListeningConnector?Operation=
GetWSDL&wsdl=SERVICE_OPERATION.VERSION
```

Use the following query format to get WSIL:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListeningConnector?Operation=
GetWSIL
```

Using Path Format to Get XML Schema, WSDL and WSIL

Use the following path format to get XML schema:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListeningConnector/
<REMOTENODE>/<OperationName>.<version>.xsd
```

Use the following path format to get WSDL:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListeningConnector/
<REMOTENODE>/<OperationName>.<version>.wsdl
```

Use the following path format to get WSIL:

```
http://<machinename>:<port>/PSIGW/PeopleSoftServiceListeningConnector/
<REMOTENODE>/inspection.wsil
```

Working With the SFTP Target Connector

This section discusses working with and setting node-level properties for the SFTP target connector.

Understanding the SFTP Target Connector

The SFTP target connector enables the gateway to use SFTP to send messages to and receive messages from SFTP servers. It uses the PUT command to place messages or files from the integration gateway onto remote SFTP servers. The GET command is used to receive messages from SFTP servers.

Target Connector Processing

The SFTP Target connector does not directly connect to SFTP servers. The connector loops back into the application server and invokes a PeopleCode application class. This application class uses the GetAttachment and PutAttachment PeopleCode functions to make the connection to the target system. At runtime, the integration gateway logs show the details of the message transfer between the application server and the integration gateway, but does not show the details of the actual connection to the SFTP server.

Base64 Encoding

The SFTP target connector features a node - level property BASE64ENCODE to specify if Base64 encoding is required for PUT requests and present for GET requests.

The following table describes the impact of this property on PUT and GET requests.

Base64 Encoding	*Description*
Y (Default.)	Base64 encoding is required for PUT requests and is present on GET requests. • Put Requests. The message body is encoded before the system sends the request from the integration gateway to the application server. The application server then decodes the request before sending it to the SFTP server. • GET Requests. The application server Base64 - encodes the data before passing it to the integration gateway. The integration gateway then decodes the response message.
N	The gateway will not Base64–encode requests or Base64–decode responses. It is the responsibility of the developer to do both.

By default the property is set to *Y.*

Most users will have no need to modify this property. This property exists to allow binary data to be passed through the connector.

When you set this property to *N,* you can Base64 - encode binary data and pass that string to the connector for a PUT request. Since the property is *N,*, and the input string is already Base64 - encode, the integration gateway has no need to perform the encoding again. The input string is then sent to the application server, which Base64 – decodes all requests being sent to the SFTP server.

Conversely, in the case of a GET request, the application server Base64 - encodes all data pulled from the SFTP server. This encoded string is then passed to the integration gateway, which decides whether to decode or not based on the value of this property. If the property is set to *N* the string is not decoded, and the system allows it to pass through as-is. You can then take the Base64 string and process it as needed.

See the “File Class Methods” (PeopleCode API Reference) product documentation for information about the GetBase64StringFromBinary and WriteBase64StringToBinary methods.

Performance Considerations

The SFTP target connector allows SFTP functionality to be available in the integration gateway using the standard target connector pattern. However, because the SFTP target connector is effectively a wrapper around PeopleCode built-in functions, there is an associated impact on performance.

In certain scenarios this inefficiency may be deemed unacceptable; in those cases call the GetAttachment and PutAttachment functions directly and bypass the integration broker entirely.

Since the gateway calls back into the application server, at least two application server processes are involved in each SFTP invocation. Please ensure that the application server is configured with the appropriate number of processes. Failure to do so may result in SFTP requests being blocked when the gateway attempts to call back into the application server.

Related Links

"Understanding the File Attachment Functions" (PeopleCode Developer's Guide)

Setting Node-Level SFTP Target Connector Properties

This section describes the required node-level properties you must set to use the SFTP target connector.

The following table describes the required node-level connector properties:

Property ID	*Property Name*	*Description*
SFTPTARGET	BASE64ENCODE	Specify if Base64 encoding is required for PUT requests or present GET requests. The values are: • *Y.* (Default.) • *N.* See the *SFTP Target Connector and Base64 Encoding* section earlier in this topic for more information about this property and the property values.
SFTPTARGET	CHARSET	Specify the character set of the data on the SFTP server. *UTF-8* is the default value. You can specify and use any of the standard character sets supported in the installed Java VM. Consult the Java docs for supported character sets. For a PUT request the system converts the string *to* the character set specified before sending it to the SFTP server. For a GET request the system converts the *from* the character set specified after it is read from the SFTP server. This property has no effect on binary data transferred when the BASE64ENCODE property is set to *N*.
SFTPTARGET	METHOD	Specifies the type of SFTP request. Values are: • *PUT.* (Default.) Push the data to the SFTP server. • *GET.* Pull the data from the SFTP server.
SFTPTARGET	REMOTEFILENAME	Specify the file name to be used on the SFTP server. For GET requests specify the name of the file to be retrieved from the SFTP server. For PUT requests specify the file name to use when sending the data to the SFTP server.

Property ID	*Property Name*	*Description*
SFTPTARGET	URL	Specify the location of the SFTP server. The URL value references a URL object stored in the system. That URL object contains the actual address of the SFTP server, as well as any user name and password required to access it. See *Understanding URL Strings Versus URL Objects* in the "Understanding the File Attachment Functions" (PeopleCode Developer's Guide) topic in the product documentation for more information..
SFTPTARGET	TEMPDIR	(Optional.) This property allows you to select the location on the application server where temporary files used during the file transfer process reside. Unless you set this property to another location, temporary files are read and written to the *PS_SERVDIR* directory. Temporary files are automatically deleted after use. If you set this property to a value other than *PS_SERVDIR,* the directory used is the concatenation of the value of the environment variable *PS_SERVDIR* and the value of the TEMPDIR property. The value of the TEMPDIR property should include any necessary slash characters needed to form a correct directory path.
SFTPTARGET	ABSOLUTEPATH	(Optional.) This property is used in conjunction with the TEMPDIR property. The values are: • *Y.* The value of the TEMPDIR property is assumed to contain the value of a complete, absolute path to a location on the application server where temporary files can be written. The value *PS_SERVDIR* is ignored. • *N.* The value of the TEMPDIR property is assumed to contain a path pointing to a directory under *PS_SERVDIR.*

To specify optional properties you must add a row to the properties grid and manually enter the property ID, property name and value.

Working With the SMTP Target Connector

This section provides an overview of the SMTP target connector and discusses how to:

- Set gateway-level connector properties.
- Set node-level connector properties.

Understanding the SMTP Target Connector

The SMTP target connector enables the gateway to send messages by email using SMTP. This connector supports plain text and HTML text content types. The connector supports the following fields: To:, From:, cc:, and bcc:. You can send data of any format in the body of the email.

You can include only one email address per type of address in the header. For instance, you can include only one addressee as a destination (DestEmailAddress).

The connector ID for the SMTP target connector is *SMTPTARGET*.

The SMTP target connect is segment-aware and you may use it to send message segments to integration partners.

See “Working With Message Segments” (Integration Broker).

Setting Gateway-Level SMTP Target Connector Properties

The SMTP target connector has one gateway-level property, in the section of the integrationGateway.properties file labeled *DELIVERED CONNECTOR CONFIGURATION Section.* This property specifies the SMTP mail server host through which the connector sends messages. Specify the host as follows:

```
ig.connector.smtptargetconnector.host=SMTP_domain_name
```

Setting Node-Level SMTP Target Connector Properties

The following table describes the required node-level properties for the SMTP target connector:

Property ID	***Property Name***	***Description***
SMTPTARGET	SourceEmailAddress	Specify the email address from which you send messages. Only one address is currently allowed.
SMTPTARGET	DestEmailAddress	Specify the email address to which you send messages. Only one address is currently allowed.
SMTPTARGET	CC	(Optional.) Specify the email address of the party to which you copy messages. Only one address is currently allowed.
SMTPTARGET	BCC	(Optional.) Specify the email address of the party to which you send blind copies of messages. Only one address is currently allowed.

Property ID	***Property Name***	***Description***
HEADER	Content-Type	(Optional.) Specify the type of text content that makes up the email body. Values are: • *Text/plain.* • *Text/html.*
HEADER	SendUncompressed	Specify whether to send messages decompressed. Values are: • *Y:* Send the message decompressed and unencoded. This is the default value. • *N:* Send the message compressed and base 64 encoded.

Chapter 6

Adding and Configuring Nodes

Understanding Nodes

Nodes represent any organization, application or system that will play a part in integrations.

For example, nodes can represent customers, business units, suppliers, other trading partners, external or third-party software systems, and so on.

Node definitions define the locations to or from which messages can be routed.

Because an application can send messages to itself, a default local node definition that represents the application is delivered as part of the integration engine.

Each PeopleSoft installation must have one, and only one, default local node

Local and Remote Nodes

Each PeopleSoft Integration Broker database involved in an integration must contain a default local node definition for itself, and a remote node definition for each of the other nodes involved.

Local and remote nodes are concepts relative to the database in which the nodes are defined. If you're signed on to Database A which has Node A defined, then Node A is local. If you're signed on to Database B, Node A is defined as remote.

For example, if the following definitions exist in the Node A database:

- NODE_A (default local)
- NODE_B (remote)

The following definitions must exist in the Node B database for it to integrate with Node A:

- NODE_A (remote)
- NODE_B (default local)

In practice, only portals use nodes designated simply as *Local*. The only local node definition used by PeopleSoft Integration Broker is the one designated *Default Local*, which represents the database onto which you are signed.

PeopleTools-Delivered Nodes

This section discusses nodes that are delivered with PeopleTools.

AIA Node

The *AIA* node is used for Oracle Application Integration Architecture (AIA) integrations, and represents an AIA integration partner.

Warning! Do not modify or delete the *AIA* node.

Anonymous Node

The *Anonymous* node is designated as the requesting node within PeopleSoft Integration Broker for third-party integrations that do not pass in a requesting node, but do have a defined any-to-local routing definition enabled on the service operation to be invoked.

Warning! Do not delete the *Anonymous* node.

You must modify the Anonymous node and define a Default User ID. The Default User ID that you specify is the ID that the system assigns to transactions that do not pass in a user ID.

Atom Node

The *Atom* node is used in association with PeopleTools feeds functionality.

You can use the *Atom* node only with asynchronous service operations. You cannot use the Atom node as the sending node. When the Atom node is the receiving node, the sending node must be the default local node.

Warning! Do not delete the *Atom* node.

Feeds are described elsewhere in product documentation.

See “Feed Publishing Framework Overview” (Feed Publishing Framework).

Default Local Node

The *Default Local Node* represents the system on which the application database is installed.

PeopleSoft Integration Broker is delivered with one node predefined as the default local node. You can't change which node is the default local node, but you can rename it to a more appropriate and meaningful name for your application or system.

Network Node

The *IB_Network* node is delivered with PeopleTools and is used to perform functionality across all nodes in the integration network, including registering network nodes on participating systems. The password set for this node must be identical on all systems participating in the network.

Warning! Do not delete the *IB_Network* node.

WSDL Node

The *WSDL* node is the default node used by the Consume Web Service wizard.

Warning! Do not modify or delete the *WSDL* node.

WADL Node

The WADL node is the default node for REST consumer services.

Warning! Do not delete the *WADL* node.

Prerequisites for Adding and Configuring Nodes

To configure a node and its associated transactions, at least one gateway with one connector must be defined.

See Defining Integration Gateways, Loading Target Connectors.

Adding Node Definitions

This section discusses how to add a node definition to the system.

Adding a Node Definition

Use the Nodes – Add a New Value page to add a node to the system.

This example illustrates the Nodes – Add a New Value page.

Note: The name you specify for a remote node must be the same as the name it specifies for itself.

To add a node:

1. Select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions**.
2. Click the Add a New Value tab.
3. In the **Node Name** field, enter a name for the node, keeping in mind that node names must begin with a character and may contain up to 30 characters.

4. Click the **Add** button to define the node.

   The Node Definitions tab displays.

Configuring Nodes

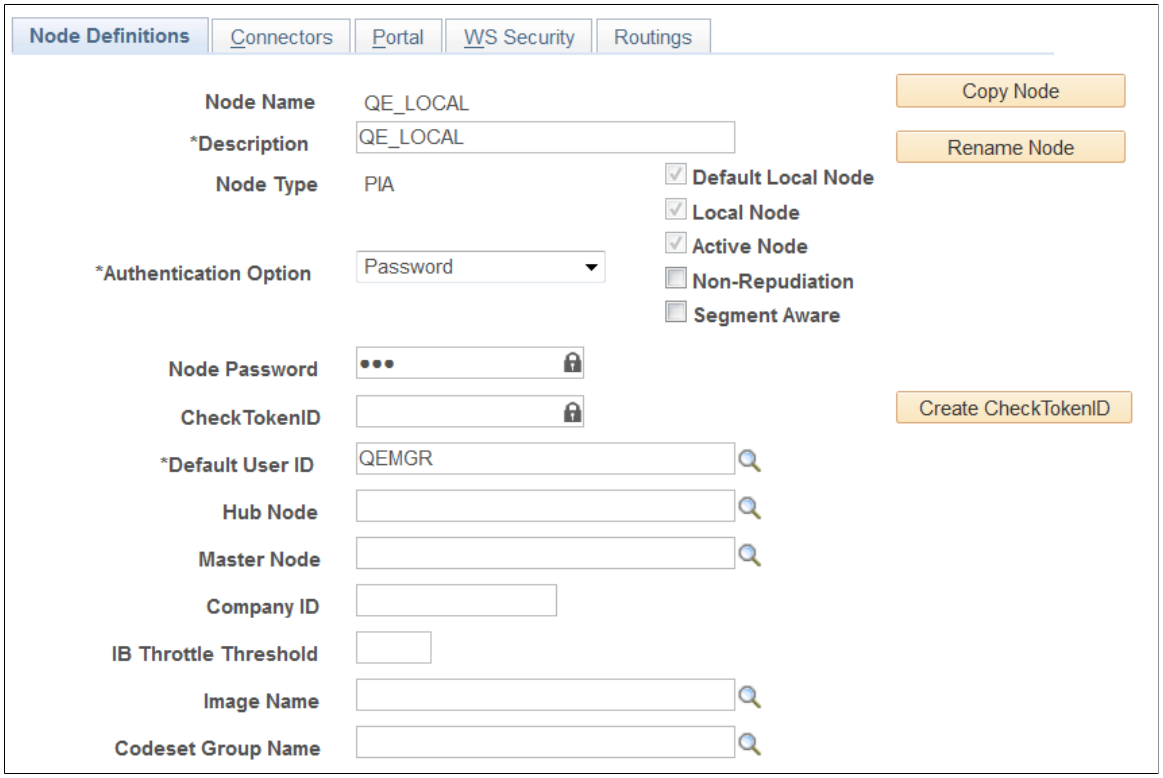
This section discusses how to:

- Define node parameters.
- Specify contact information.
- Define node properties.
- Specify node gateways and connectors.

Defining Node Parameters

Access the Node Definitions page (**PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions**.)

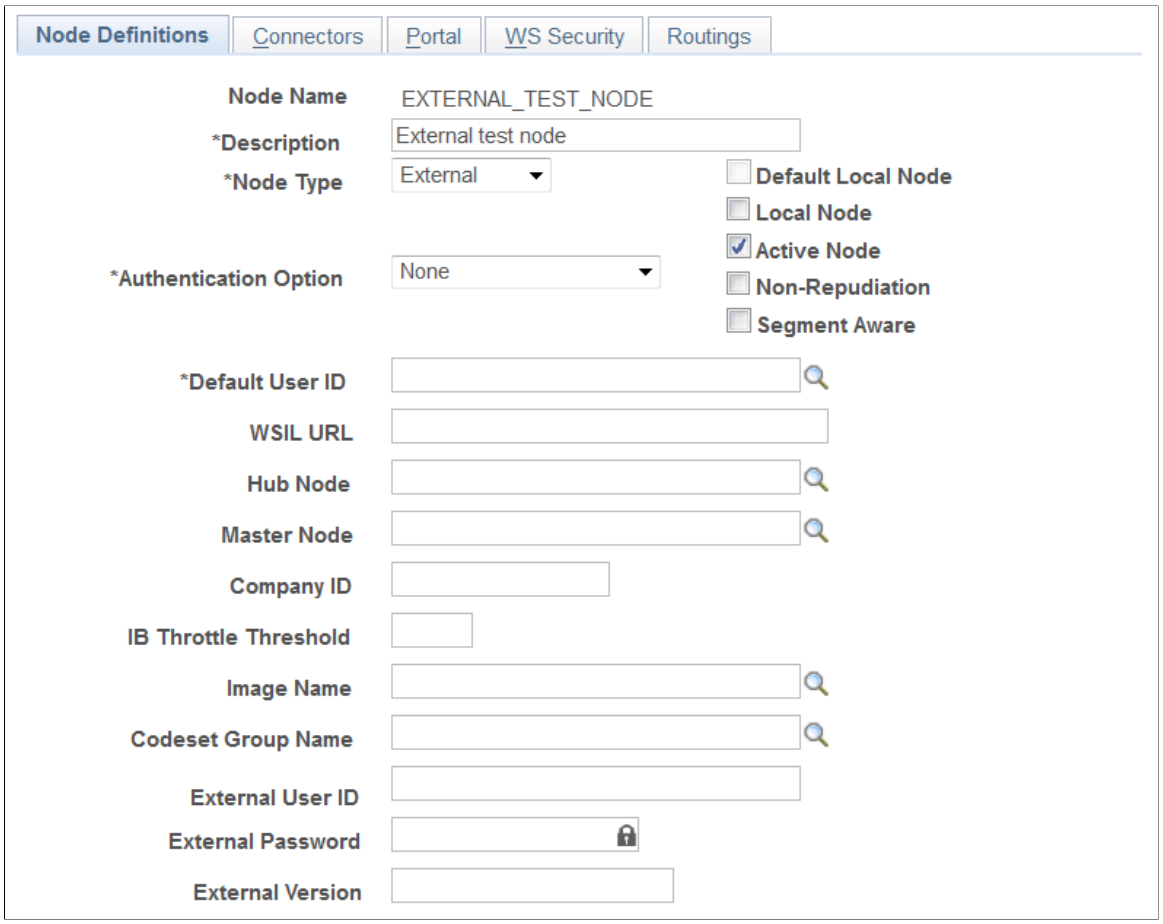
This example illustrates the fields and controls on the Nodes - Node Definitions page for the default local node. You can find definitions for the fields and controls later on this page.



The previous example shows the node definition for the node *QE_LOCAL*, the default local node.

When the definition is for an *External* type node, additional fields appear on the page.

This example illustrates the fields and controls on the Nodes - Node Definitions page for an external node. You can find definitions for the fields and controls later on this page.



The previous example shows the node definition for the node *EXTERNAL_TEST_NODE,* an *External* type node.

Field or Control	***Description***
Description	Enter a descriptive name for the node.
Node Type	Select from: *PIA:* Designates the node as a PeopleSoft database that uses PeopleSoft Integration Broker. This is the default for a new node. *External:* Designates the node as an entity that doesn't use PeopleSoft Integration Broker. *ICType:* A portal-specific setting that PeopleSoft Integration Broker doesn't use.

Field or Control	*Description*
Authentication Option	Select from: • *Certificate*: The current node uses a digital certificate to sign the messages it sends, and expects messages it receives to be signed by a complementary digital certificate. When a PeopleSoft Pure Internet Architecture node receives a service operation, PeopleSoft Integration Broker extracts the distinguished name from the certificate and validates it against the sending node's distinguished name retrieved from the default local node's keystore. Service operations sent by the default local node have the digital certificate automatically inserted by Integration Broker. An external node is expected to respond to certificates outwardly the same way as a PeopleSoft Pure Internet Architecture node. • *None:* No authentication is required. This is the default value. ***Warning!*** Single signon is not compatible with this option. If you select *None* for the default local node, and implement single signon on the same system, all transactions will fail. You must select either *Password* or *Certificate* when implementing single signon. • *Password*: Two new fields appear:**Password** and **Confirm Password.** Enter your password in the first edit box, and confirm it in the second edit box. With a PeopleSoft Pure Internet Architecture node, PeopleSoft Integration Broker expects service operations, both outbound to and inbound from the current node, to include a password, which it validates against the password entered here. An external node is expected to respond to passwords outwardly the same way as a PeopleSoft Pure Internet Architecture node. See Implementing Node Authentication.
Default Local Node	Indicates whether the current node represents the database to which you are assigned.
Local Node	Indicates that the current node is either a portal node or the default local node. You cannot change this setting for the default local node.

Field or Control	*Description*
Active Node	Select to make the current node definition active, so it can be used by PeopleSoft Integration Broker. Clear the box to inactivate the node. Note the following points about inactivating a node: • You cannot inactivate the default local node. • Inactivating a node will inactivate related routing definitions. You must reactivate the routing definitions manually. See “Activating and Inactivating Routing Definitions” (Integration Broker).
Non-Repudiation	Select to activate nonrepudiation for the current node. Note that to activate nonrepudiation for the current node you must also activate nonrepudiation in the service operation definition for which you want this feature.
Segment Aware	Check the box to configure the node to handle message segments. See “Working With Message Segments” (Integration Broker).
Node Password	Displays when the **Authentication Option** is *Password*. Enter a node password. The limit for this field is 88 characters.
Confirm Password	Reenter the node password you entered in the **Password** field.
Create CheckTokenID	This control appears only when the **Authentication Option** type is *Password* or *Certificate.* Click the button to create a system-generated check tokenID for use in conjunction with single signon among PeopleSoft systems. When you click the button, the system populates the **Check TokenID** field with the generated value. See “Implementing PeopleSoft-Only Single Signon” (Security Administration) for more information about check token IDs.

Field or Control	*Description*
Check TokenID	This field appears only when the **Authentication Option** type is *Password* or *Certificate.* This field contains the check token ID value generated after clicking the **Create CheckTokenID** button. This ID is used in conjunction with single signon among PeopleSoft systems. Alternatively, you can create a custom ID up to 256 characters. **Note:** Copy the value in the field before saving the page. You must provide this value to other PeopleSoft partners/nodes participating in single signon, as they must define this value on their database on the remote node definition that represents your database. After you save the page, the field becomes masked. See “Implementing PeopleSoft-Only Single Signon” (Security Administration) for more information about check token IDs.
Default User ID	On inbound integrations, this is the user ID that the sender must specify to invoke a service operation, unless you have set up an external user ID for this purpose. On outbound integrations, this is the default user ID sent with the service operation.
WSIL URL	This field appears only when working with an *External* node type. This field is used in conjunction with using introspection to create routing definitions. Enter the WSIL URL for the target system to include in the routing definition.
Hub Node	Select the name of a node that will serve as a “gatekeeper” for the current node. You can select any existing PeopleSoft Pure Internet Architecture node for this purpose. Not all node types are appropriate as hub nodes. Nodes of type ICType are portal-specific, and aren’t used by PeopleSoft Integration Broker. A node of type External typically isn’t an Integration Broker system, so it might not be usable as a hub node unless you’ve explicitly configured it to be compatible with Integration Broker.
Master Node	This field is for information only. If the current node is used as a hub, you can indicate the target node with which it’s associated. If the current node represents a subordinate database, you can indicate the primary database.
Company ID	Enter the name of the company or organization associated with the current node.

Field or Control	*Description*
IB Throttle Threshold	Set this parameter on a remote node definition to limit the number of requests sent to the node per dispatch. The setting is in minutes. For slow-processing systems, this option can help to prevent saturating the targeting system with requests. This parameter is used only for asynchronous integrations.
Image Name	Select an image from the system database. Any application that uses images can use the selected image to represent the current node.
Code Set Group Name	Select the codeset group to which you want the current node to belong. Transform programs invoked by service operations use this association to search for message data requiring translation. See “Performing Data Translation” (Integration Broker).
Copy Node	The **Copy Node** button displays after you have saved the initial node definition. Click to define a new node with the same properties as the current node. The Default Local check box is cleared for all new nodes. **Note:** If you copy a local node, the new node will be local as well. You must clear the Local Node check box to use it with PeopleSoft Integration Broker.
Rename Node	The **Rename Node** button displays after you have saved the initial node definition. You can rename only the default local node. Additional information about deleting nodes is contained elsewhere in this topic. See Renaming or Deleting Nodes.
Delete Node	The **Delete Node** button displays after you have saved the initial node definition. You cannot delete the default local node. Additional information about deleting nodes is contained elsewhere in this topic. See Renaming or Deleting Nodes.

Field or Control	Description
External ID	This field appears only when working with an *External* node type. This field is used for outbound integrations in conjunction with implementing WS-security. See the *Specifying External User IDs and Password* section in the Implementing Web Services Security topic.
External Password	This field appears only when working with an *External* node type. This field is used for outbound integrations in conjunction with implementing WS-security. See the *Specifying External User IDs and Password* section in the Implementing Web Services Security topic.
External Version	This field appears only when working with an *External* node type. This field is currently not used.

Specifying Contact Information

Click the Contacts/Notes link to access the Node−Node Contacts/Notes page.

Each node represents a database or other software entity managed by one or more people. Use this page to record information about the people associated with the current node.

Field or Control	Description
Contact Manager	The name of the representative or administrator of the node, in standard PeopleSoft name format.
Contact Email	The Contact Manager’s email address, in standard PeopleSoft email address format.
Contact Phone Number	The phone number of the contact manager.
Contact URL	The address of the Contact Manager’s support web site, if there is one.

Defining Node Properties

Click the Properties link to access the Node − Properties page.

This page provides a convenient place to store additional information about the current node that can be referenced by any other node. Properties created for all nodes are stored in a single table, PSNODEPROP.

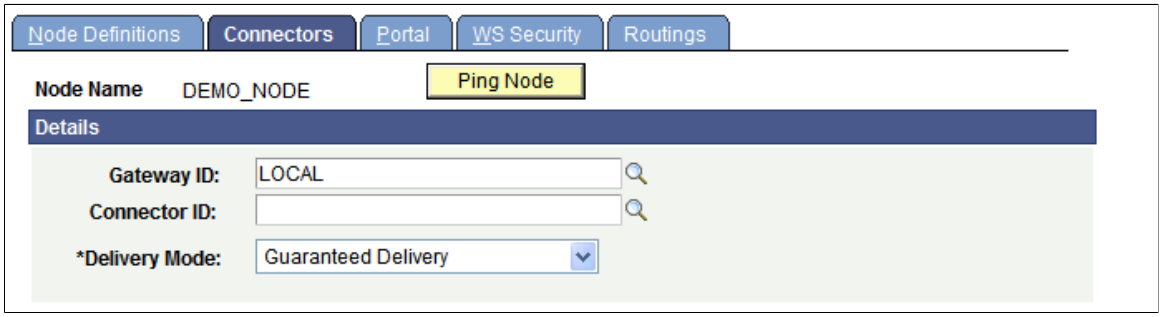
Examples include a DUNS number or Tax Identification Number. These properties can be used to update messages with additional information. They can also serve to add additional categorization for custom processing; for example, add a Region property so nodes can be referenced by region for special processing.

Field or Control	***Description***
Name Type	Select from: *Category:* The property is used for categorization. *Ident:* The property is used for identification. *Search:* The property is used for searching.
Property Name	Enter a new property name or select an existing property of the selected name type.

Specifying Gateways and Connectors

Select the Connectors tab to access the Nodes − Connectors page.

This example illustrates the Nodes – Node Connectors page.



Use this page to specify the integration gateway and target connector the node uses for integrations.

At least one gateway with at least one target connector must be defined and configured. If the current node is remote, it can use the default local node's gateway or any other installed gateway as its local gateway. If the current node has its own gateway installed, the default local node's database must contain a definition for it, configured as a remote gateway.

Specifying a Gateway and Target Connector for the Current Node

To specify a gateway and connector for the current node:

1. From the Gateway ID field, click the Lookup button to select the gateway ID for the gateway you want the current node to use.

When the default local node sends a message to any other node, the message first goes to the default local node's local gateway through its PeopleSoft listening connector, regardless of the gateway ID you select here.

- If you specify a remote gateway ID, the local gateway uses its default remote gateway connector (specified in the integrationGateway.properties file) to route messages to the remote gateway through the remote gateway's PeopleSoft listening connector. The remote gateway sends the messages directly to the current node, using the connector you specify in the next step.

  Note: The default remote gateway connector setting initially specifies the HTTP target connector, which is unlikely to change unless you develop a custom target connector.

- If you specify the local gateway ID, the local gateway sends messages directly to the current node, using the connector you specify in the next step.

2. From the Connector ID field, select a connector ID from the list of connectors registered with the selected gateway.

   Specify the target connector appropriate to the communication method preferred by the current node. If the node is a PeopleSoft application with Integration Broker installed, select *PSFTTARGET*. If the node is a PeopleSoft 8.1x application, select *PSFT81TARGET*.

   The rows on the Properties and Data Type/Description tabs are automatically populated with the connector's properties that are designated *Required* in the gateway definition. The fields on these tabs are the same as those on the Connector Properties page. If the connector has multiple instances of a required property defined, only the instance designated as *Default* appears.

   See Editing Connector Properties.

3. From the Delivery Mode drop-down list box, select a delivery mode. The options are:

   - Guaranteed. (Default.)
   - Best Effort.

   These options are discussed in detail elsewhere in this PeopleBook.

   See Setting Target Connector Delivery Modes.

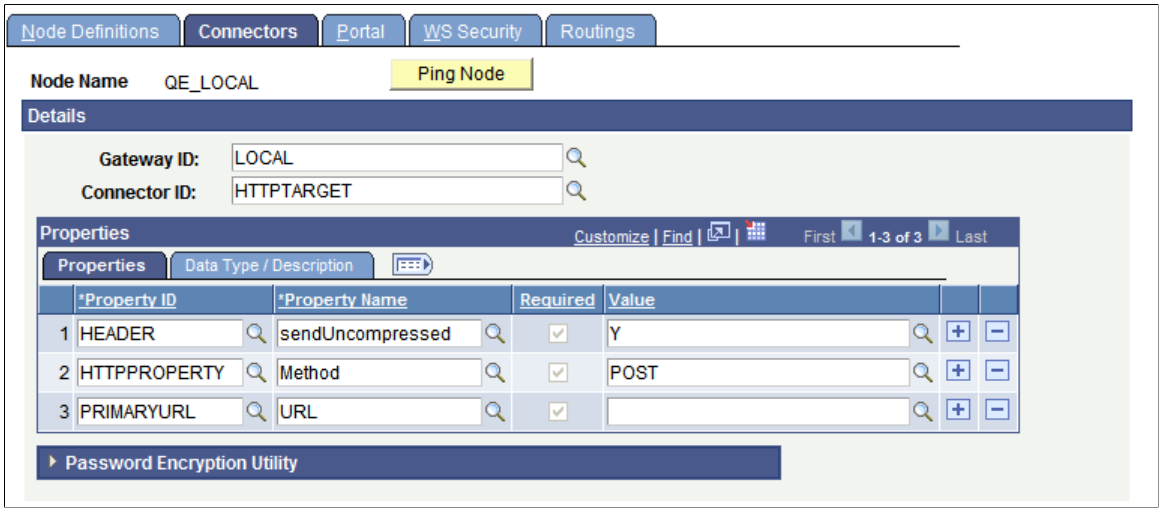
4. Click the **Save** button.

Note: You can override the gateway and connector selection for individual outbound transactions.

Working With Connector Properties

Properties that appear on the Nodes – Connectors page are copies of the specified connector's required properties.

This example illustrates the Node – Connectors page with the HTTP target connector defined. The default properties for this connector appear in the Properties grid.



You can use this page to:

- Add an instance of a non-required property.
- Add a new instance of a required property.
- Modify the value or description of a property instance.
- Remove a property instance.

Information about appropriate modifications might come from PeopleSoft, from the connector's developer, or from your own experience and requirements.

Important! Don't remove a required property unless you replace it with another instance of the same property. Without all of its required properties, the connector is unlikely to work correctly.

You must encrypt any password connector property values. The Connector tab features access to the Password Encryption Utility that enables you to encrypt a password value and paste it into the appropriate field on the page. To access the utility, click the **Password Encryption Utility** arrow.

Accessing the Integration Gateway Properties File

The Connectors tab features a **Gateway Setup Properties** link you can use to access the integrationGateway.properties file directly from this tab.

Testing Connector Configurations

The Connectors tab features a **Ping Node** button you can use to test your configuration.

If the ping is successful, a window displays with a message indicating that the gateway is active and the PeopleTools version that you are running. If the ping is not successful, a window displays with a message indicating the gateway could not be found.

Related Links

Editing Connector Properties

Pinging Nodes

This section discusses how to:

- Ping a node using the Nodes–Connectors page.
- Ping a node using the Node Status page.

Understanding Pinging Nodes

This section describes the processing that takes place when you ping a node from the PeopleSoft Pure Internet Architecture.

- The system uses the application server URL, user ID and password specified in the Jolt connect string settings section in the integration gateway properties file to establish a connection to the target database.
- The system verifies that the user ID and password are valid on the target system.

  The destination node must match the defined local node on the target system, otherwise the system displays the following error:

  ```
  Destination node does not match local node.
  ```

  When pinging remote nodes, if the source node defined on the target is defined as a PeopleSoft node and an authentication token is passed, then the authentication option on the node must be either *Node Password* or *Certificate.* The system then compares the passwords or certificate.

  In the following cases, if node authentication is defined, you cannot ping a node:

  - The source node defined on the target system is not a PeopleSoft node.
  - The source node defined on the target system is a PeopleSoft node, but no authentication option, for example *Node Password* or *Certificate,* is selected.

Pinging Nodes Using the Nodes-Connectors Page

You can ping a node using the Nodes-Connector page in PeopleSoft Integration Broker.

To ping a node:

1. Access the Nodes-Connectors page (select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** and click the **Connectors** tab).
2. Click the **Ping Node** button.

Pinging Nodes Using Node Status Page

You can ping a node using the Node Status page in the Service Operations Monitor.

To ping a node:

1. Access the Node Status page (select **PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Node Status**).
2. Click the **Ping Node** button.

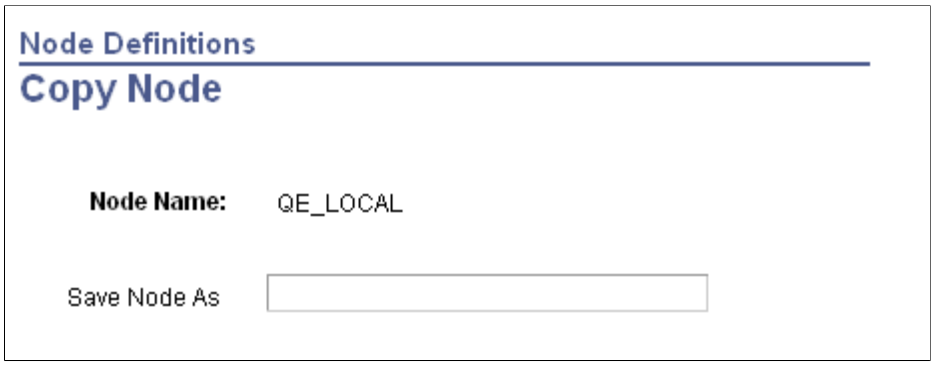
Related Links

Pinging Nodes

Copying Nodes

To define a new node with the same properties as an existing node, use the Copy Node page (IB_NODE_SAVEAS). To access the page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** and click the **Copy Node** button.

This example illustrates the fields and controls on the Copy Node page. You can find definitions for the fields and controls later on this page.



If you copy the default local node, the **Default Local** option is cleared on the new node.

The following fields and controls appear on the page:

Field or Control	*Description*
Node Name	Name of the node to copy.
Save Node As	Name of the new node.
Save	Click the button to save the changes.
Cancel	Click the button to exit the page without saving the changes.

To copy a node definition:

1. Open the node definition to copy.
   a. Select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions.**
   b. Click the name of the node to copy.

      The Nodes – Node Definition page appears.
2. Click the **Copy Node** button.

   The Copy Node page appears.
3. In the **Save Node As** field enter the name of the new node.
4. Click the **Save** button.

   The definition for the new node appears in the Nodes – Node Definition page.

Renaming or Deleting Nodes

This section discusses how to rename and delete nodes.

Understanding Renaming and Deleting Nodes

This section discusses how to rename and delete nodes.

There are several situations in which you might need to rename or delete a node definition. When you do so, PeopleSoft Integration Broker automatically handles most of the dependencies involved — such as deleting routings and other properties associated with the node.

However, the live message data in Integration Broker Monitor remains unchanged. If that data still contains references to the node you want to modify, Integration Broker will prevent you from making the modification. You must remove all data from the live message tables before you can rename or delete the node definition.

You cannot delete the default local node or a node that hosts a portal. As a result, the Delete Node button is hidden on these node definitions.

Note: If you upgraded your PeopleSoft application from a PeopleTools 8.1x release, the newly created default local node definition must be renamed, so you must first remove any remaining live message data if you didn’t do so before the upgrade.

Renaming or Deleting a Node

Renaming or deleting a node requires the following actions:

1. Deactivate all the domains in your messaging system.
   a. Access the Domain Status page.

b. For each active domain in the system, from the **Domain** drop-down list box, select Inactive.

c. Click **Update** to change the status of all domains to *Inactive* and all dispatchers to *Cleanup*.

d. Click **Force Reset** to change the status of all dispatchers to *Inactive*.

2. Remove the data from the live message tables.

   You have several choices when removing data from the live message tables:

   - You can archive messages one at a time from the Asynchronous Details or Synchronous Details component.
   - You can archive messages with a batch process using the Archive Monitor Data Archive Monitor Data component.
   - You can purge message data using one of several Data Mover scripts delivered with PeopleSoft Integration Broker. You'll find them in *PS_HOME*\scripts:

Field or Control	***Description***
AppMsgPurgeLive.dms	Deletes the queue data from every live message table in the database.
AppMsgPurgeAll.dms	Deletes the message data from every live message table and every archive message table in the database. This is the recommended procedure when upgrading from earlier versions of PeopleTools, because the archived data is largely incompatible with the new release.

3. Rename or delete the desired node definition.

   If you are renaming the default local node, note that the name cannot exceed 15 characters. Other node names can contain up to 30 characters.

4. Reboot the web server.

5. Reactivate the messaging domains.

   a. Access the Domain status page.

   b. On the Domain Status page, select **All Domains Active**.

   c. Click **Update** to change the status of all domains and dispatchers to *Active*.

Related Links

“Understanding Running Batch Service Operation Archiving Processes” (Integration Broker Service Operations Monitor)

“Purging Runtime Monitor Tables” (Integration Broker Service Operations Monitor)

“Understanding Data Mover Scripts” (Lifecycle Management Guide)

Chapter 7

Configuring PeopleSoft Integration Broker for Handling Services

Understanding Configuring PeopleSoft Integration Broker for Handling Services

This section provides an overview of several of the service configuration properties that you must set to use services with PeopleSoft Integration Broker.

Namespaces

Namespaces provide a method for qualifying element and attribute names that are used in XML documents and are identified by Uniform Resource Identifier (URI) references.

To *provide* PeopleSoft services you must specify a service namespace and a schema namespace.

The Service Configuration page enables you to define default values for these namespaces. You may redefine the default values for specific objects when needed:

- You can redefine the service namespace for each PeopleSoft service definition
- You can modify namespaces in each nonrowset-based message's schema.

You cannot change the direction of a REST service (provider or consumer) once defined in the PeopleSoft system. However, with respect to a non-REST service, a simple change to the routing definition can change its direction from consumer to provider and vice versa.

As a result it’s good practice to set the default values for the service namespace and schema namespace, regardless of whether your system is a service provider or a service consumer of REST or non-REST services.

Target Locations

When configuring the PeopleSoft system for services, you set target locations for web services and for REST services, using the Target Locations page.

For web services, target locations are URLs that PeopleSoft Integration Broker uses to build and validate XML message schemas, export WSDL documents, and as the SOAP endpoint. For REST services, target locations are URLs that PeopleSoft Integration Broker uses to export WADL documents and as the REST endpoint.

PeopleSoft Integration Broker enables you to define an unsecured target location and a secured target location URL for each type of service, and provides separate fields for doing so.

In general, the URL you specify as a target location should be an unsecured URL.

Note: XML schema validation will fail if you enter a secured URL in the Target Location field.

If you need to use a secure URL for a SOAP or REST endpoint, you may do. Any secure URL you enter overrides any unsecured URL entered.

In addition, you can override the (unsecured) target location URL for exporting WSDL and WADL documents on a case–by–case basis. The Provide Web Service wizard features a Use Secure Target Location box in Step 2 – Select Service Operations of the Wizard. If you select the box, the system exports the WSDL or WADL document to the URL specified in the secure target location defined on the Target Locations page. The override is in effect only for that particular export of WSDL or WADL. If you need to generate WSDL or WADL again for the service, the system defaults back to using the (unsecured) target location URL, unless you again check the Use Secure Target Location box in the wizard to override the URL to use the secure target location.

To use services with PeopleSoft Integration Broker, you must specify web service and REST service target locations. Specifying a secure target location for either of these is optional..

Service System Status

The Services Configuration page contains a **Service System Status** drop-down list box that enables you to restrict rename, delete, and other administrative actions that users can perform on services, service operations, messages, and other integration metadata.

You can select one of two values from the drop-down list box: *Production* or *Development*.

The following table describes the impact of each of these setting on managing integration metadata:

Object	***Action***	***Production Mode***	***Development Mode***
Messages	Rename	You cannot rename a message that is associated to a service that has WSDL or WADL provided. You must first delete the WSDL or WADL documents before you can rename the message.	An alert message displays indicating that WSDL or WADL documents have been provided for the service to which the message is associated, but you may continue with the action and rename the message.
Messages	Delete	You cannot delete a message that is associated to a service that has WSDL or WADL provided. You must first delete the WSDL or WADL documents before you can delete the message.	An alert message displays indicating that WSDL or WADL documents have been provided for the service to which the message is associated, but you may continue with the action and delete the message.

Object	*Action*	*Production Mode*	*Development Mode*
Message Schemas	Delete	You cannot delete a message schema that is associated to a service that has WSDL or WADL provided. You must first delete any WSDL or WADL documents before you can rename the schema.	An alert message displays indicating that WSDL or WADL documents have been provided for the service to which the message schema is associated, but you may continue with the action and rename the schema
Queues	Rename	The Service System Status has no impact on renaming queue definitions. However, you cannot rename a queue if it is referenced in a service operation or if it is referenced in the runtime tables.	The information that applies to renaming queues in production mode also applies to renaming queues in development mode.
Queues	Delete	The Service System Status has no impact on deleting queue definitions. However, you cannot delete a queue if it is referenced in a service operations or if it is referenced in a runtime table.	The information that applies to deleting queues in production mode also applies to deleting queues in development mode.
Routings	Rename	The Service System Status has no impact on renaming routing definitions.	The information that applies to renaming routing definitions in production mode also applies to renaming routings in development mode.
Routings	Delete	You cannot delete an any-to-local routing definition that is tied to a service that has WSDL or WADL provided. You must first delete the WSDL or WADL document from the service before deleting the routing definition.	An alert message displays indicating that WSDL or WADL documents have been provided for the service to which the routing is associated, but you may continue with the action and delete the routing definition.
Service	Rename	You cannot rename services that have had WSDL or WADL documents provided. The WSDL or WADL documents must be deleted before you can rename a service.	An alert message displays indicating that WSDL or WADL documents have been provided for the service, but you can continue with the action and rename the service.

Object	*Action*	*Production Mode*	*Development Mode*
Service	Delete	The Service System Status has no impact on deleting services. However, you cannot delete any service that is referenced by a service operation.	The information that applies to deleting services in production mode also applies to deleting services in development mode.
Service Operation	Rename	You cannot rename service operations that are associated to services that have WSDL or WADL provided. You must delete the WSDL or WADL before you can rename the service operation.	An alert message displays indicating that WSDL or WADL documents have been provided for the associated service, but you may continue with the action and rename the service operation.
Service Operation	Delete	• You cannot delete service operations that are associated to services that have WSDL or WADL provided. You must delete the WSDL or WADL document before you can delete the service operation. • If you delete the default service operation version, all versions of the service operation are deleted. You cannot delete a service operation that is referenced in the runtime tables.	An alert message displays indicating that WSDL or WADL documents have been provided for the associated service, but you may continue with the action and delete the service operation. You cannot delete a service operation that is referenced in the runtime tables.
Service Operation	Change Service	You cannot change a service operation that is associated to a service that has WSDL or WADL provided. You must first delete the WSDL or WADL documents before you can modify the setting. The new service to which you associate an operation may have had WSDL or WADL generated. However, if you want the WSDL or WADL from the newly associated service operation to be included in the WSDL or WADL document, you must export the service again.	An alert message displays indicating that WSDL or WADL documents have been provided for the associated service, but you may continue with the action and change the service associated with the service operation. The new service to which you associate an operation may have had WSDL or WADL generated. However, if you want the WSDL or WADL from the newly associated service operation to be included in the WSDL or WADL document, you must export the service again.

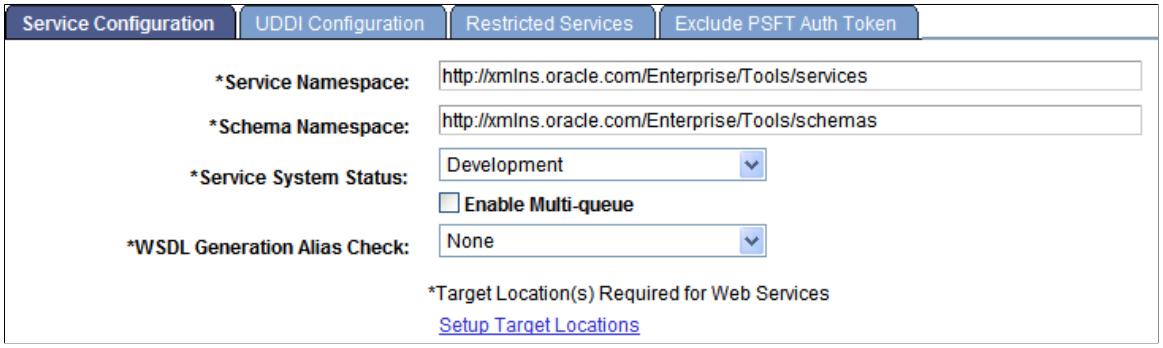
Using the Service Configuration Page to Set Service Configuration Properties

You set service configuration properties on the Service Configuration page (IB_SVCSETUP) and on the Target Locations page (IB_SVCSETUP_SEC) in the Services Configuration component (IB_SVCSETUP).

Use the Service Configuration page to define the service namespace, schema namespace and service system status. Use the page to also enable multi-queue functionality whereby multiple queues process asynchronous transactions, and to set alias enforcement in generated WSDL.

To access the Service Configuration page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration.**

This example illustrates the Services Configuration – Service Configuration page.



Using the Target Locations Page to Set Target Locations for Services

Use the Target Locations page (IB_SVCSETUP_SEC) to set service target locations. To access the page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the **Setup Target Locations** link.

This example illustrates the Target Locations page.

Use the Web Services Target Locations section of the Target Locations page to set an unsecured or secured URL to be used for XML message schema, WSDL, and as the SOAP endpoint when working with SOAP-based services. Use the REST Services Target Location section of the Target Locations page to set an unsecured or secured URL to be used for WADL and as the REST endpoint.

Important! Target locations for both Web services and REST services must be set for the system to be properly configured to handle services.

Use the example code provided as a guide for entering target locations.

Note: The REST target location URL must include the local default node.

Setting Service and Schema Namespaces

To set service and schema namespaces:

1. Access the Service Configuration Properties page (**PeopleTools** > **Integration Broker** > **Configuration** > **Services Configuration**).
2. In the **Service Namespace** field, declare a service namespace.
3. In the **Schema Namespace** field, declare a schema namespace.
4. Click the **Save** button.

Related Links

Namespaces
Using the Target Locations Page to Set Target Locations for Services

Setting Service System Status

To set the service system status:

1. Access the Service Configuration Properties page (**PeopleTools** > **Integration Broker** > **Configuration** > **Services Configuration**).
2. From the **Service System Status** drop-down list box, select one of the following options:
   - *Development.* (Default.)
   - *Production*.

   These statuses are discussed elsewhere in this topic

   See Understanding Configuring PeopleSoft Integration Broker for Handling Services.
3. Click the Save button.

Related Links

Using the Target Locations Page to Set Target Locations for Services

Setting the System for Multi-Queue Processing

To set the system for multi-queue processing:

1. Access the Service Configuration Properties page (**PeopleTools** > **Integration Broker** > **Configuration** > **Services Configuration**).
2. (Optional) Check the **Enable Multi-queue** box to use multiple queues to process inbound and outbound asynchronous requests.

   This feature is discussed elsewhere in the product documentation.

   See “Assigning Multiple Queues to Process Service Operations” (Integration Broker).
3. Click the **Save** button.

Enabling WSDL Alias Generation Checking

To enable WSDL alias generation checking:

1. Access the Service Configuration Properties page (**PeopleTools** > **Integration Broker** > **Configuration** > **Services Configuration**).
2. From the **WSDL Generation Alias Check** drop-down list, select an option for enforcing field and record alias names in generated WSDL.

   This feature is discussed elsewhere in the product documentation.

See “Enforcing Message Record and Field Aliases in Generated WSDL” (Integration Broker).

Setting Target Locations for Services

To set the target location for services:

1. Access Target Location page (**PeopleTools** > **Integration Broker** > **Configuration** > **Services Configuration** and click the Set Target Locations link.).
2. Locate the Web Services Target Locations section of the page.
3. In the **Target Location** field, enter an unsecured URL to be used for XML message schema, WSDL, and as the SOAP endpoint.

   If you enter a secure URL in the **Target Location** field, XML message schema validation will fail.

   If you require a secure target location for the SOAP endpoint, see Step 4.

   If you have a dedicated integration gateway, the format of the value that you enter is:

   ```
   http://<machine>:<port>/PSIGW/PeopleSoftServiceListeningConnector
   ```

   If the default local node points to a different gateway server where WSDL documents and XSD schemas are available, use the alternate location URL format: .

   ```
   http://<machine>:<port>/PSIGW/PeopleSoftServiceListeningConnector
   /<defaultlocalnode>
   ```

4. (Optional) In the **Secure Target Location** field, enter a secure URL to be used as the SOAP endpoint.

   The URL entered here overrides the target location defined in Step 3 for the target location for the SOAP endpoint.

   The URL you enter here is also used as a secure target location for exporting WSDL, if you choose the **Use Secure Target Location** box in the Provide Web Service wizard.

   If you do not enter a value in this field, the system uses the unsecured URL that you specify in the **Target Location** field for both the SOAP endpoint and for exporting WSDL.

   See “Step 2: Select Service Operations” (Integration Broker) for providing Non-REST web services, “Step 2: Select Service Operations” (Integration Broker) for providing REST web services, and “Step 2: Select Service Operations” (Integration Broker) for providing OpenAPI REST web services..

   If you have a dedicated integration gateway, the format of the value that you enter is:

   ```
   https://<machine>:<port>/PSIGW/PeopleSoftServiceListeningConnector
   ```

   If the default local node points to a different gateway server, use the alternate location URL format:

   ```
   https://<machine>:<port>/PSIGW/PeopleSoftServiceListeningConnector
   /<defaultlocalnode>
   ```

5. Click the **OK** button.

   The Service Configuration page appears.

6. Click the **Save** button.

Related Links

Target Locations
Using the Target Locations Page to Set Target Locations for Services

Setting Target Locations for REST Services

To set the target location for REST services:

1. Access Target Location page **PeopleTools** > **Integration Broker** > **Configuration** > **Services Configuration** and click the Set Target Locations link.).
2. Locate the REST Services Target Location section of the page.
3. In the **Target Location** field, enter an unsecured URL to be used when exporting WADL documents and as the REST endpoint.

   If you require a secure target location for the REST endpoint, see Step 4.

   The format of the value that you enter is:

   ```
   http://<machine>:<port>/PSIGW/RESTListeningConnector/<defaultlocalnode>
   ```

4. (Optional) In the **Secure Target Location** field, enter a secure URL to be used as the REST endpoint.

   The URL entered here overrides the target location defined in Step 3 for the target location for the REST endpoint.

   The URL you enter here is also used as a secure target location for exporting WADL, if you choose the **Use Secure Target Location** box in the Provide Web Service wizard.

   If you do not enter a value in this field, the system uses the unsecured URL that you specify in the Target Location field for both the REST endpoint and for exporting WADL.

   See “Step 2: Select Service Operations” (Integration Broker) for providing REST web services and “Step 2: Select Service Operations” (Integration Broker) for providing OpenAPI REST web services.

   If you have a dedicated integration gateway, the format of the value that you enter is:

   ```
   https://<machine>:<port>/PSIGW/RESTListeningConnector/<defaultlocalnode>
   ```

5. Click the **OK** button.

   The Service Configuration page appears.

6. Click the **Save** button.

Related Links

Target Locations
Using the Target Locations Page to Set Target Locations for Services

Chapter 8

Specifying UDDI Repositories in PeopleSoft Systems for Providing and Consuming Services

Understanding Specifying UDDI Repositories in PeopleSoft Systems

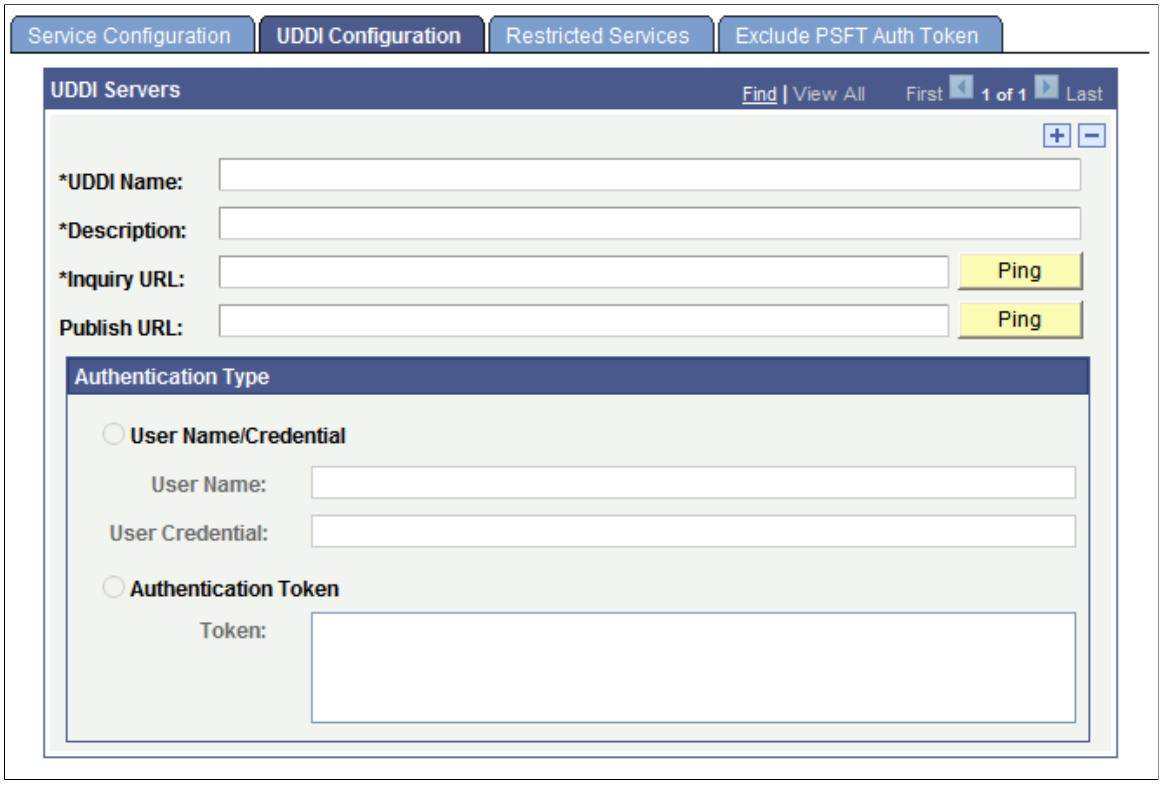
You can provide services to and consume services from one or more UDDI repositories. Before doing so, you must configure each repository in the PeopleSoft system.

Specifying UDDI Repositories in the PeopleSoft System

Use the Service Configuration-UDDI Configuration page (IB_SVCSETUP2) to specify UDDI repositories in the PeopleSoft system for providing services to and consuming services from UDDI repositories.

To access this page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the **UDDI Configuration** tab.

This example illustrates the Services Configuration – UDDI Configuration page.



To specify a UDDI repository in the PeopleSoft system:

1. Access the UDDI Configuration page (**PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the **UDDI Configuration** tab).
2. In the **UDDI Name** field, enter the name of the UDDI server.
3. In the **Description** field, enter a descriptive information for the UDDI server.
4. Specify the URL used when consuming services from UDDI repositories.
    a. In the **Inquiry URL** field, enter the URL to use to inquire for services available on the UDDI server. This is the URL used when consuming services from UDDI repositories. It is also used when publishing to UDDI repositories to inquire the server for possible existing WSDL document versions.
    b. Click the **Ping** button next to the **Inquiry URL** field to verify that you entered the correct URL.
5. Specify the **Publish URL**.
    a. In the **Publish URL** field, enter the URL for publishing WSDL documents to the UDDI server. This URL is used when providing services to UDDI repositories.
    b. Click the **Ping** button next to the Publish URL field to verify that you entered the correct URL.

To specify additional UDDI repositories to use for providing or consuming services, click the plus (+) button at the top right corner of the UDDI Server section to add a row.

Chapter 9

Managing Pub/Sub Server Domains

Understanding Managing Pub/Sub Domains

PeopleSoft Integration Broker includes a set of Oracle Tuxedo servers that monitor database tables and process items in the tables. The processing can include running PeopleCode programs, creating publication and subscription contracts, and so forth.

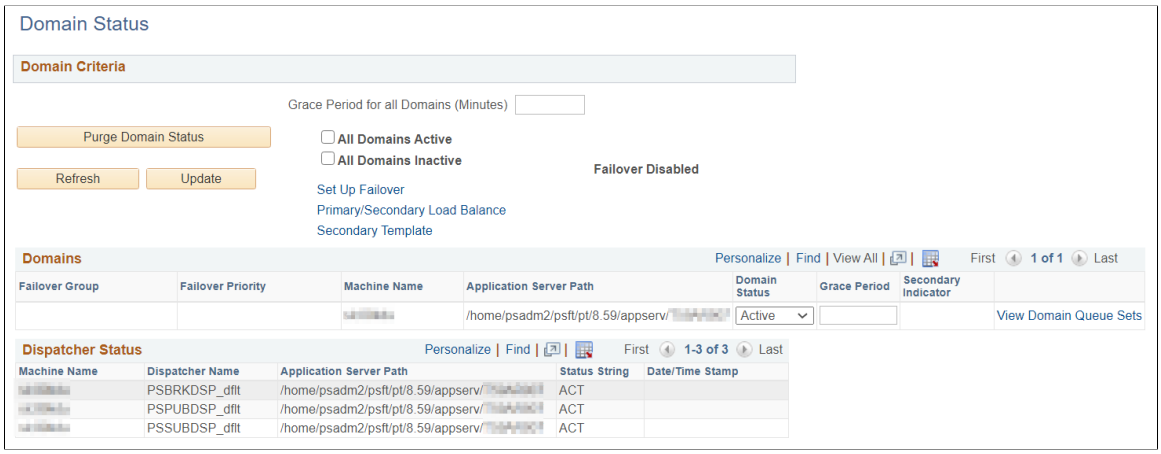
The Domain Status page enables you to view the domains that have pub/sub servers on them that are running against the application database. You can also use this page to manually set domain grace periods to allow processing in a domain to finish before you pause the processing or take the domain offline.

In addition, if a machine with a domain on it crashes, the integration system may still operate as if the processes in the domain are still working on items in the runtime tables. The Domain Status page enables you to set the domains to inactive so that other pub/sub servers can complete the processing of these items. This enables you to recover from domain and machine crashes.

Working with the Domain Status Page

The Domain Status page (AMM_MULTIDOM) features three sections, the Domain Criteria section, the Domain Status section, and the Dispatcher Status section.

This example illustrates the fields and controls on the Domain Status page. You can find definitions for the fields and controls later on this page.



The Domain Criteria section enables you to perform actions on all domains in the integration system, such as apply a grace period to all domains, activate or inactivate all domains, and purge the current information in the Dispatcher Status section.

The Domains section enables you to activate and inactivate domain status and set domain grace periods. You can also use this section to view failover information for a domain.

The Domain Status section provides application server name and path information for all machines that have domains on the messaging system. For any machine, you can use the drop-down list box to activate or inactivate the machine and all domains on it. You can also set grace periods for domains on specific machines.

Note: If the domain is in Production mode as defined on the Service Configuration page, then the Integration Broker domain status value is set to Active by default.

See Using the Service Configuration Page to Set Service Configuration Properties.

The Domain Status page also features the following controls:

Field or Control	***Description***
Purge Domain Status	Click the button to purge all of the current status information in the Dispatcher Status section. After you click this button, the system populates the section with information about all processes that are still running. **Note:** Purging domain status purges all domains, including the domain on which the PeopleSoft Pure Internet Architecture (PIA) is running. Click the Refresh button to refresh information about the domain running PIA. The rate at which the system re-registers information for the domain running PIA depends on the scan interval setting in PSADMIN.
Update	Click the button to save or apply changes that you make in the Domain Criteria section or the Domain Status section.
Force Reset	Click the button to reset the status of all entries in the Dispatcher Status column in the Dispatcher Status section to *Inactive*.
Refresh	Click the button to refresh the Domains section and Dispatcher Status section of the page.

Viewing Dispatcher Status

The Dispatcher Status section of the Domain Status page displays information about machines in the integration system that have dispatcher processes associated with them. This area displays the machine name, the dispatcher process name, the application server path, the dispatcher status, and any grace periods set for a process running on the domain.

There are three valid dispatcher status values:

Field or Control	*Description*
ACT	Indicates that the dispatcher process is active on the domain.
INACT	Indicates that the dispatcher process is inactive on the domain. No processing occurs.
CLNUP	Indicates that the dispatcher process is in clean-up mode. The pub/sub server releases queued items for processing and waits for items currently processing to finish. The time that appears in the grace period column indicates when the cleanup process will end. The time equals the system time and the clean up time interval that you enter.

Activating Pub/Sub Server Domains

Before you can use the pub/sub system, you must activate the domain on which a pub/sub server resides.

To activate a domain:

1. Select **PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status.**.

   The Domain Status page appears.

2. In the Domains section:

   a. Locate the row that lists the machine where the domain resides that you want to activate.

   b. In the **Domain Status** drop-down list box, select *Active.*

3. Click the **Update** button.

Inactivating Pub/Sub Server Domains

To inactivate pub/sub servers on domains:

1. Inactivate pub/sub server domains:

   a. To inactivate domains on all machines in the messaging system, select the **All Domains Inactive** check box. To activate the servers at a later time, select the **All Domains Active** box.

   b. To inactivate domains on individual machines, locate the domains to inactivate. In the drop-down list box, select *Inactivate*. To activate the servers at a later time, select *Activate* in the list.

2. Click the **Update** button.

The domain status for the domains that you inactivate changes from *Active* to *Inactive.* In addition, in the Dispatcher Status section, the dispatcher status of all processes associated with the domains changes from active (*ACT*) to cleanup (*CLNUP*). Click the **Refresh** button until the dispatcher status changes to inactive (*INACT*).

If you inactivated all domains, a **Force Reset** button appears under the Update button. The **Force Reset** button enables you to force the dispatcher status to change from cleanup to inactive.

Changing Dispatcher Status for Processes

The **Force Reset** button appears only when you change the domain status for all domains on all machines by selecting the All Domains Inactive check box.

To change dispatcher status for all processes on all machines from cleanup to inactive:

1. Click the **Force Reset** button.
2. Click **Update**.

Setting Domain Grace Periods

The time that appears in the **Grace Period** column indicates when the cleanup process ends. The time equals the system time and the cleanup time interval that you enter.

To set one grace period to apply to domains on all machines, locate the **Grace Period for all Domains** field in the Domain Criteria section and enter the number of minutes for the grace period. Click **Update**.

To set grace periods for individual domains, enter the number of minutes for the grace period for each domain. Click **Update**.

A grace period that you set for an individual domain takes precedence over the setting for all groups.

The grace period setting for all domains is a convenient way to set a grace period for all dispatchers in all the domains. You can set a grace period of all domains at the top of the page and then press the **Tab** key to access individual domains and override the group setting.

Chapter 10

Using the Integration Network WorkCenter

Understanding the Integration Network WorkCenter

PeopleSoft Integration Broker features an Integration Network WorkCenter that is delivered with the following pagelets:

Term	*Definition*
Integration Network pagelet	The Integration Network centralizes the administrative tasks of configuring, administering, and monitoring integrations in PeopleSoft Integration Broker. This feature also enables you to create a network of PeopleSoft integration partner systems and view system configuration status, verify integration processing, and more on those systems. The integration network enables you to activate and deploy groups of integrations, assess the security permission status of service operations, and more. The integration network also provides the ability to monitor integration activity in the network by node or by transaction.
IB Configure Network pagelet	The IB Configure Network pagelet is an activity guide that leads you through the process of performing a basic configuration of PeopleSoft Integration Broker and an integration network. This activity guide is intended for those with little experience setting up and using PeopleSoft Integration Broker. It is also intended for those with a requirement to configure PeopleSoft Integration Broker to use other PeopleTools technologies, such as the PeopleSoft Test Framework, PeopleSoft Search, Feeds, ADS, and other technologies.

Related Links

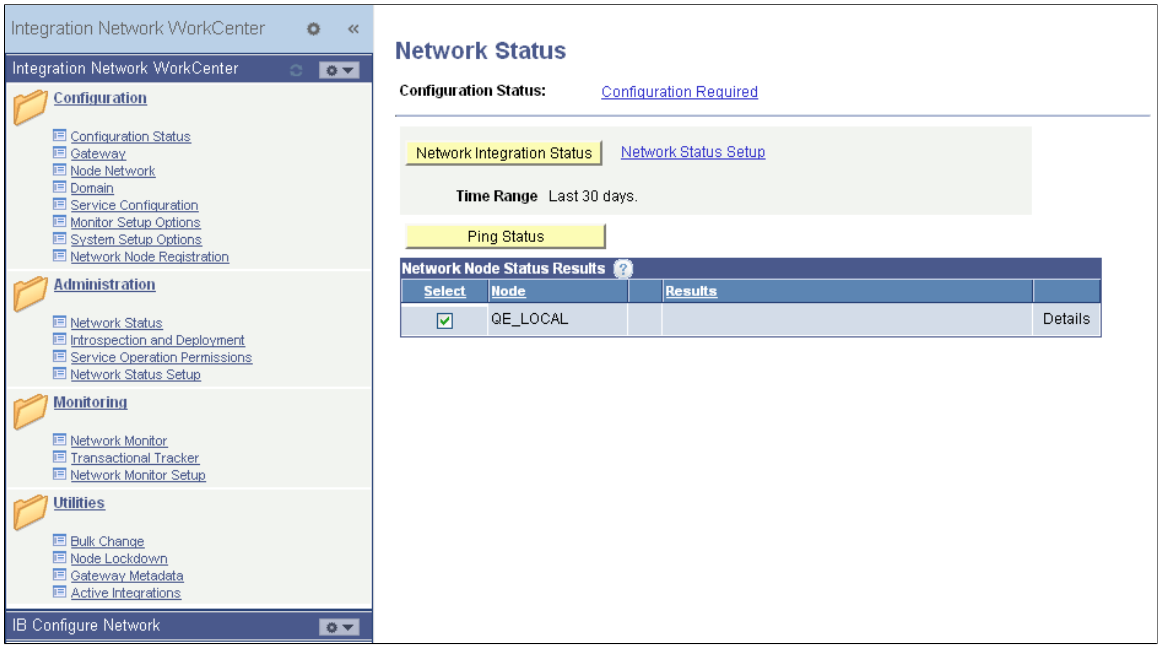
Understanding the Integration Network

Understanding the Integration Broker Configuration Activity Guide

Accessing the Integration Network WorkCenter

To access the Integration Network WorkCenter select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter.**

This example illustrates the default view of the Integration Network WorkCenter



By default, the Integration Network pagelet appears when you access the Integration Network WorkCenter. To access the IB Configure Network pagelet and the Integration Broker Configuration activity guide, minimize the Integration Network pagelet and expand the IB Configure Network pagelet.

Use the following control to minimize and expand pagelets in the Integration Network WorkCenter:

Field or Control	***Description***
	The button is an expand/minimze toggle button. Click the button to expand a pagelet and click the button again to minimize a pagelet.

Information about using and customizing WorkCenters is provided elsewhere in the product documentation. See the product documentation for Applications User's Guide.

Related Links

Accessing the Integration Network

Accessing and Navigating the Integration Broker Configuration Activity Guide

Chapter 11

Using the Integration Network

Understanding the Integration Network

The integration network centralizes the administrative tasks of configuring, administering, and monitoring integrations in PeopleSoft Integration Broker. The network is delivered as a pagelet in the Integration Network WorkCenter.

The integration network provides configuration and node status pages that allow you to quickly assess if the integration network is properly configured, as well as identify the nodes that are in the network and their configuration status. The integration network enables you to activate and deploy groups of integrations, assess the security permission status of service operations, and more. The integration network also provides the ability to monitor integration activity in the network by node or by transaction.

Important! Only nodes defined as PeopleSoft nodes (*PIA* node type) can participate in the integration network.

Accessing the Integration Network

This section discusses how to access the integration network using the Integration Network WorkCenter.

Understanding Accessing the Integration Network

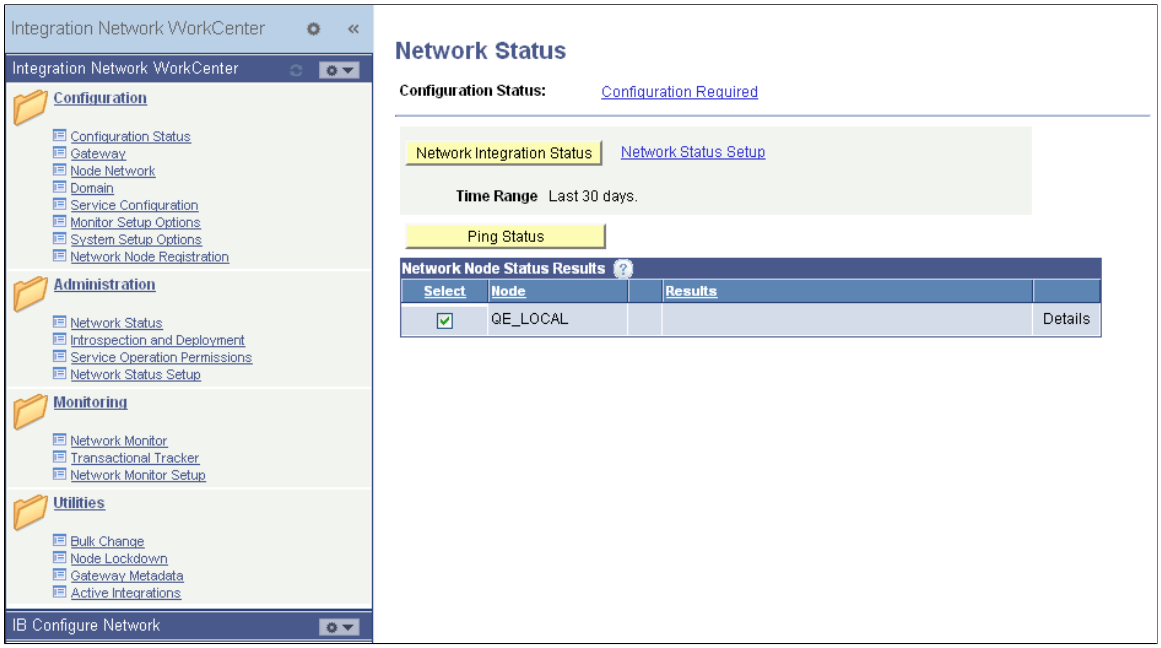
You can access integration network features using the Integration Network WorkCenter which is located in the PeopleSoft Pure Internet Architecture.

Note: Procedures for accessing and using the integration network are documented in the product documentation using the Integration Network WorkCenter pagelet.

Accessing the Integration Network Using the Integration Network WorkCenter

To access the Integration Network WorkCenter , select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter.**

This example illustrates the default view of the Integration Network WorkCenter.



By default, the Integration Network WorkCenter is delivered with two pagelets: the Integration Network WorkCenter and the IB Configure Network activity guide. The Integration Network WorkCenter pagelet appears expanded. The IB Configure Network activity guide is described elsewhere in the product documentation.

Click the **Expand** button on a pagelet to view a pagelet. Click the **Minimize** button to collapse a pagelet.

The left navigation pane of the Integration Network WorkCenter provides links to the integration network features and pages. When you select a link from the left navigation pane, the corresponding page appears in the main work area. By default, the Network Status page appears in the main work area.

Information for working with and customizing WorkCenters is provided elsewhere in PeopleTools product documentation.

See the product documentation for *PeopleTools 8.55: Using PeopleSoft Applications*.

Configuring the Integration Network

This section discusses how to:

- Use the Configuration Status page.
- Verify and manage integration gateway configuration.
- Verify node network status.
- Add and modify nodes in the PeopleSoft database.
- Add nodes to the network.
- Verify publication/subscription server domain status.

- Set the network node password.

Understanding Configuring the Integration Network

The integration network is properly configured when the integration gateway is configured, when the default local node is configured, local and remote nodes are configured, and when the application server domain is active. The Integration Network WorkCenter features a Configuration Status page that enables you to quickly assess if the integration network is configured. The page also provides access to the PeopleSoft Integration Broker components used to perform configuration tasks if you want to view or modify settings.

Understanding Minimum Configuration Requirements for the Integration Network

The Configuration Status page indicates if the integration network is minimally configured to run. The following table lists the minimum configuration requirements for the network.

Note: All nodes to be managed and monitored in the Integration Network must be PeopleSoft nodes. PeopleSoft nodes are those nodes defined as *PIA* node types in the node definition.

Required Configuration	***Page and Object ID***	***Integration WorkCenter Navigation***	***Alternate Integration Broker Navigation***

• Define a local gateway. This gateway represents the source system.
  • In the definition you must specify a gateway URL and specify that the gateway is local.
  • Load the target connectors.
• Define one or more remote gateways. These gateways represent the target systems.
  • In each remote gateway definition you must specify the URL of the remote gateway.
  • Load the target connector for each gateway.
	Gateways – Gateways page
	IB_GATEWAY
		Gateway Configured
		• Update Gateway Location
		• Register Target Connectors
			PeopleTools > **Integration Broker** > **Configuration** > **Integration Gateways.**

Required Configuration	*Page and Object ID*	*Integration WorkCenter Navigation*	*Alternate Integration Broker Navigation*
Register and configure PeopleSoft nodes on the local gateway. You must first create local and remote node definitions before you can register and configured them on the local gateway.	Gateways – PeopleSoft Node Configuration page (PSGTWPROPS_SEC)	Gateway Configured, Update Configuration Settings	**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways.** > **Gateway Setup Properties.**
In the Integration Gateway CERTIFICATE Section of the integration gateway properties file for each gateway defined, specify the secure keystore path and password.	Gateway Properties page (BGWPROPERTIESPAGE)	Node Network Configured	**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways.** > **Gateway Setup Properties** > **Advanced Properties Page.**
Default Local Node: • Rename or define the local default node. This node represents the source system. • Activate the default local node. Remote Nodes: • Define remote nodes as warranted. The remote nodes represent target systems. • Activate the nodes. • Assign the nodes to a gateway. After you create local and remote node definitions, you must register and configure them on the gateway using the PeopleSoft Node Configuration page.	Nodes – Node Definitions (IB_NODE)	Node Network Configured	**PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** > **Node Definitions.**
Assign the default local node and any other local nodes to a gateway and target connector.	Nodes-Connectors page (IB_NODECONN)	Node Network Configured	**PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** > **Connectors.**

Required Configuration	*Page and Object ID*	*Integration WorkCenter Navigation*	*Alternate Integration Broker Navigation*
Activate the application server domain.	Domain Status (AMM_MULTIDOM)	Domain Active	**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status.**
Set the network node password.	Update Network Node Password (IB_INTNET_NODEPWD)	Node Network Password (This link is located in the Additional Links section at the bottom of the page.)	**PeopleTools** > **Integration Broker** > **Integration Network** > **Configuration Status** . In the Additional Links section, click **Node Network Password.**

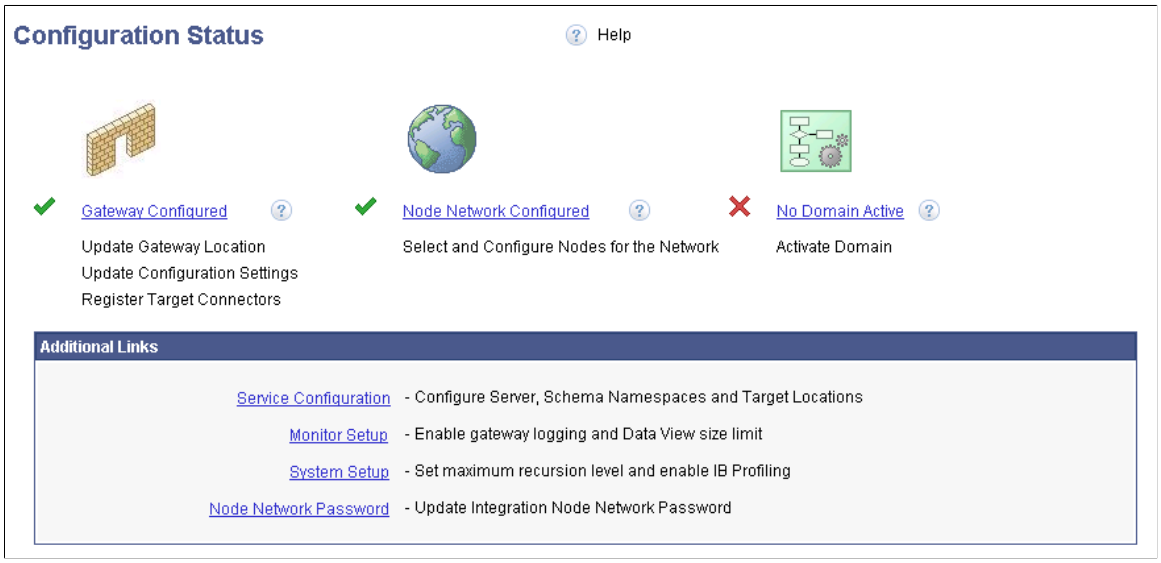
Using the Configuration Status Page

The Configuration Status page indicates if the integration network is minimally configured to run as described in the previous section.

The page provides visual cues that enable you to quickly asses if the integration network is properly configured. Integration components that are properly configured display on the page with a green check mark icon next to them; components that are not configured, are not configured properly, or that require additional configuration display on the page with a red “X” icon next to them.

To access the Configuration Status page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration** > **Configuration Status.**

This example illustrates the fields and controls on the Configuration Status page. You can find definitions for the fields and controls later on this page.



In the previous example, green check mark icons appear next to the **Gateway Configured** and **Node Network Configured** links and indicate that each of these integration components are properly

configured. A red “X? icon appears next to the Activate Domain link, and indicates that the domain is not active.

The following page elements appear on the Configuration Status page:

Field or Control	***Description***
	A green check mark icon appears next to an integration network component to indicate that the component is properly configured.
	A red “X” icon appears next to an integration network component to indicate that the component is not configured, requires additional configuration, or is not properly configured.
Gateway Configured	Click the link to access the Gateways component to view the gateway configuration or to configure the gateway. For the integration network to be properly configured, the integration gateway must be configured.
Node Network Configured	Click the link to access the Node Network page and view, configure and manage PeopleSoft nodes in the integration network. For the integration network to be properly configured, the node network must be configured.
Domain Active	Click the link to access the Domain Status page in the Service Operations Monitor to activate an application server domain. For the integration network to be properly configured, the application server domain must be active.
Service Configuration	(Optional.) Click the link to access the Service Configuration page where you specify services settings, such as the service namespace, the schema namespace, the target location, and so on.
Monitor Setup Options	(Optional.) Click the link to access the Monitor Setup Options page to set display and other options in the Service Operations Monitor. For example, use the page to verify or set the proper size limit (Data Length View Limit) for displaying XML in the Service Operations Monitor for asynchronous transactions, enable gateway logging, and more.
System Setup Options	(Optional.) Click the link to access the System Setup Options page to set the recursion level (Message Builder Depth Limit) for message parts, enabling runtime profiling to generate system performance statistics, and more.

Field or Control	*Description*
Node Network Password	Click the link to access the Update Node Network Password page where you specify the node network password. The network node is used to perform functionality across all nodes in the integration network, including registering network nodes on participating systems.

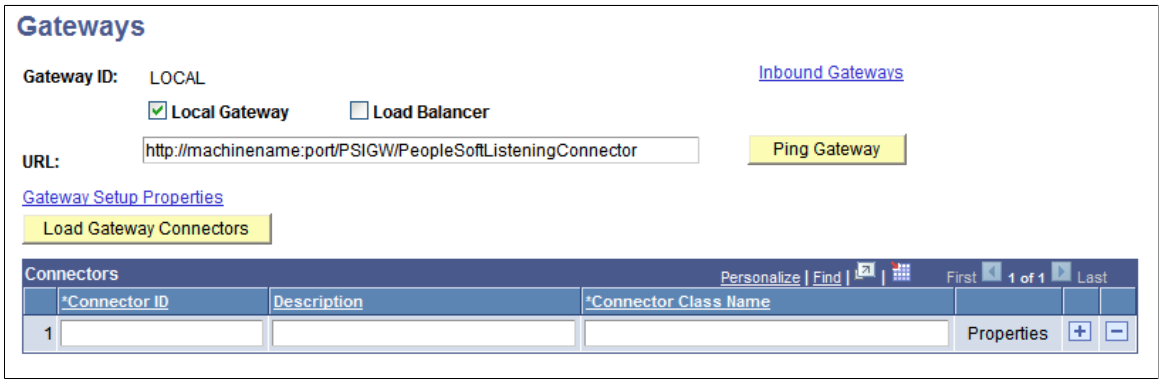
Verifying and Managing Integration Gateway Configuration

At a minimum the following settings and options must be configured on the integration gateway for the integration network to function properly:

- Define the default gateway URL.
- Load the delivered target connectors.
- Register nodes.
- Define the gateway keystore path and keystore password.

The Configuration Status page features a Gateway Configured link. When you click the link the Gateways component appears and you can view and configure integration gateway settings and options. You can also use the Gateway link in the Integration Network WorkCenter navigation pane to access the Gateways page and the features described in this section.

This example illustrates the Gateways page when no integration gateway has yet been defined:



This table provides links to documentation that describes how to perform the minimum tasks required to configure the integration network:

Task	*Documentation*	*Comments*
Define local and remote gateways.	See Defining Integration Gateways.	NA
Load target connectors for each gateway.	See Administering Integration Gateways	See the topic “Loading Target Connectors, Loading Connectors by Introspection.

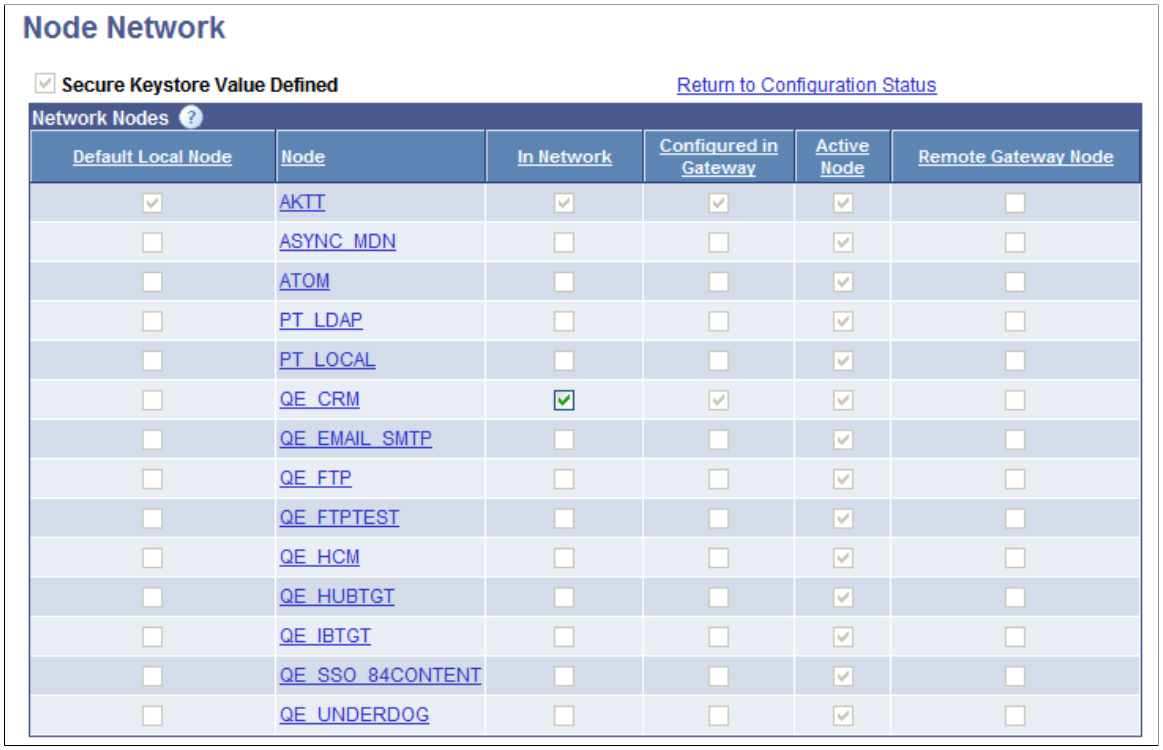
Task	*Documentation*	*Comments*
Register nodes.	See Setting Oracle Jolt Connection Properties.	You must create the node definitions before you can register and configure them on integration gateways.
Define the gateway keystore path and password.	See Configuring Security and General Properties	Gateway digital certificates must be installed on the gateway. ***Warning!*** Integrations will fail if you do not set the secure keystore path and password.

Viewing Node Network Status

The Configuration Status page features a **Node Network Configured** link that opens the Node Network page. Use the Node Network page (IB_INTNETWORK) to view and configure details about PeopleSoft nodes in the integration network. The Node Network page provides a view of all the PeopleSoft nodes in the database.

Note: Only PeopleSoft nodes can be viewed and managed in the integration network pages. A PeopleSoft node is one defined as a *PIA* node type on the Nodes – Node Definition page.

This example illustrates the fields and controls on the Node Network page. You can find definitions for the fields and controls later on this page.



The Node Network page allows you to easily identify the current default local node, nodes that are defined in the integration network, active nodes, and so on.

The following page elements appear on the Node Network page:

Field or Control	***Description***
Secure Keystore Value Defined	This page element is read-only. A check mark in the box denotes that the secure keystore path and password has been defined in the integration properties file for the default local gateway. The Node Network component is not set to *Configured* until the box is checked; The box displays checked after you define the keystore path and password in the integrationGateway. properties file. ***Warning!*** Integrations will fail if you do not set the secure keystore path and password. See Configuring Security and General Properties
Default Local Node	This page element is read-only. A check mark in this column denotes the default local node in the system. There can be only one default local node in the PeopleSoft database. Additional information about the default local node is provided elsewhere in the product documentation. See Adding and Modifying Nodes in the PeopleSoft DatabaseUnderstanding Nodes.
Node	The node column lists the PeopleSoft nodes in the database. Only nodes defined as PeopleSoft nodes appear on the page. PeopleSoft nodes are those defined as *PIA* node types in the node definition. Information about adding and modifying node definitions is provided elsewhere in this topic and in the PeopleTools product documentation. See Adding and Modifying Nodes in the PeopleSoft DatabaseUnderstanding Nodes.

Field or Control	*Description*
In Network	A check mark in this column denotes that the node is defined as part of integration network. **Important!** A node must be “in network? for you to be able view integration information and use the features of the integration network on the node. The default local node is automatically defined in the network. To include a node in the integration network, check the box. To remove a node from the integration network, clear the box. The In Network box is enabled only when the node is active and you have defined jolt connection strings for the node in the integration gateway. Additional information about adding nodes to the integration network is provided elsewhere in this topic. See Registering Nodes in the Network.
Configured in Gateway	This page element is read-only. A check mark in this column denotes that the jolt connection strings for the node have been defined in the integration gateway. Defining jolt connection strings for nodes is described elsewhere the product documentation. See Setting Oracle Jolt Connection Properties.
Active Node	This page element is read-only. A check mark in this column denotes that the node had been defined as an active node in the node definition. To change the active status of a node, click the node name in the grid to access the node definition. Information about activating a node is described elsewhere in the product documentation. See Defining Node Parameters.
Remote Gateway Node	This page element is read-only. A check mark in this column denotes that the node is defined on a remote gateway. If the node is defined using a remote gateway, the defined gateway will not be checked to determine if the node information is configured in that remote gateway.

Adding and Modifying Nodes in the PeopleSoft Database

You can click the name of any node in the Node Network list to open the node definition in the Node Definition page and view or modify the node definition.

The following table lists the actions you must take to properly configure a node for the node network. The table also provides links to the corresponding product documentation that describes how to perform the actions:

Action	*Documentation*	*Comments*

• Define the default local node.
• Define a remote node for each additional system participating in integrations.
	See Understanding Nodes Prerequisites for Adding and Configuring Nodes, Adding Node Definitions, Configuring Nodes, Specifying Gateways and Connectors.	• You can rename the delivered default local node or define a new one.
		• For each node use the Nodes-Node Definition page to verify the following:
		  • The node is defined as a *PIA* node type.
		  • The node is set to *Active.*
		• For each node use the Nodes-Connectors page to assign the node to an integration gateway and target connector.

Registering Nodes in the Network

This section describes how to:

- Use the Node Network page to register nodes in the network.
- Use the Network Node Registration page to register nodes in the network.

Understanding Registering Nodes in the Network

A node must be registered in the network for you to send and receive integrations with other network integration partners and to use many of the features of the integration network on the node.

You can view nodes that are registered in the integration network using the Node Network page, shown earlier in this topic. A check mark in the In Network column on the Node Network page indicates that the node is defined as part of the integration network.

Note: The default local node is automatically registered as a network node.

You can use two pages to register a node in the integration network. This table describes the differences between the two options:

Field or Control	*Description*
Network Node page	To register a node in the network, you click the **In Network** page control. For a node to be available to register using this page it must be: • Registered in the integration gateway. • Defined as an active node in the local database.
Network Node Registration page	For a node to be available to register using this page it must be registered in the integration gateway. **Important!** Local and remote nodes must be using PeopleTools 8.53 or higher integration gateways to use the Network Node Registration page to register nodes in the network. The processes on this page perform the following actions on the local database: • Create an active node definition for the remote node. • Register the node in the integration network. • If applicable, register the locally defined default local node with the integration gateway. The node information must be in the PSIBNODEREG table, typically populated as part of OVM. The processes on this page perform the following actions on the remote database: • Create an active definition for your local node. • Register the node in the network. Node definitions created using this page use the same authentication option that is defined on the local default node's system.

Prerequisites for Registering Nodes in the Network

The prerequisites for registering nodes in the integration network vary, depending on the page that you use.

To register a node in the integration network using the Network Node page the following conditions must exist.

- The node must be active.

Information about activating a node is described elsewhere in the product documentation.

See Defining Node Parameters.

- The jolt connection strings for the node must be defined in the integration gateway

  Information about setting the jolt connection strings for a node is described elsewhere in the product documentation.

  See Setting Oracle Jolt Connection Properties.

To register a node in the integration network using the Network Node Registration page the following conditions must exist:

- The local and remote nodes must be using PeopleTools 8.53 or higher integration gateways.
- The node must be defined in the local gateway on the PeopleSoft Node Configuration page.

  See Setting Oracle Jolt Connection Properties

- The network node password must be set on the local and remote systems.

  Information for setting the network node password is provided

- To create the locally-defined node on remote network databases the authentication option defined on the IB_NETWORK node on the local database *must* be the same as defined on each remote database where the node is to be created. Before attempting to create the local node on remote databases ensure that the authentication options are properly updated on the IB_NETWORK node.

Using the Node Network Page to Register Nodes in the Network

To register a node in the integration network using the Node Network page (IB_INTNETWORK):

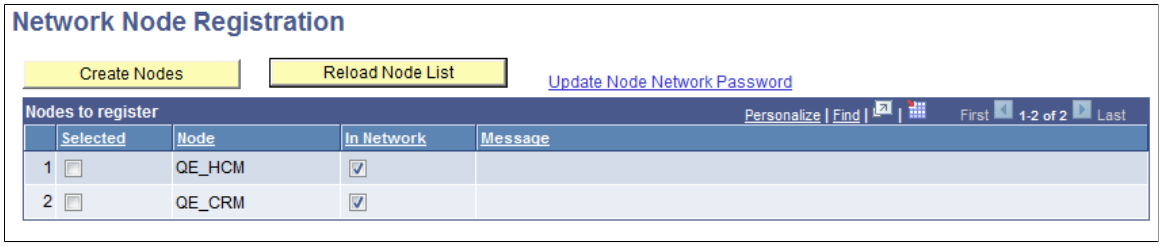
1. Access the Network Node page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration** > **Node Network.)**
2. In the In Network column, check the box next to the node that you want to register.
3. Click the **Save** button.

Using the Network Node Registration Page to Register Nodes in the Network

This section describes using the Network Node Registration page (IB_INTNET_NODEREG) to register nodes in the integration network.

To access the Network Node Registration page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration** > **Register Network Nodes.**

This example illustrates the fields and controls on the Network Node Registration page. You can find definitions for the fields and controls later on this page.



The page displays all nodes defined in the integration gateway on the PeopleSoft Node Configuration page that are not currently configured in the local database.

The following fields and controls appear on the page:

Field or Control	***Description***
Create Nodes	Click the control to perform the following tasks on selected nodes: • Create node definitions on the local database. • Create a node definition for the local node on the remote databases. • Register the selected nodes in the integration network as in-network nodes. • If applicable, register the locally-defined default local node with the integration gateway. The node information must be in the PSIBNODEREG table (typically populated as part of OVM).
Reload Node List	Click the control to refresh the page with nodes that are not currently defined in the local database.
Update Node Network Password	Click the link to access the Update Node Network Password page to specify the network node password.
Selected	Select the box to include a node in the registration process.
Node	Node name.
In Network	Check the box to register a node in the network. By default this control is selected.
Message	Displays the results of the node registration.

To register a node in the network using the Network Node Registration page:

1. Access the Network Node Registration page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration** > **Register Network Nodes**.)
2. In the Nodes to Register grid, check the **Selected** box next to each node to include in the registration process.
3. Click the **Create Nodes** button.

   The results of the registration process appear in the Message field for each selected node.

Verifying Publication/Subscription Server Domain Status

For the integration network to be properly configured the pub/sub server domain must be set to Active.

When the **No Domain Active** link appears on the Configuration Status page, the pub/sub server domain is not set to Active. Click the link to access the Domain Status page in the Service Operations Monitor and to activate the domain status.

See Activating Pub/Sub Server Domains.

Setting the Network Node Password

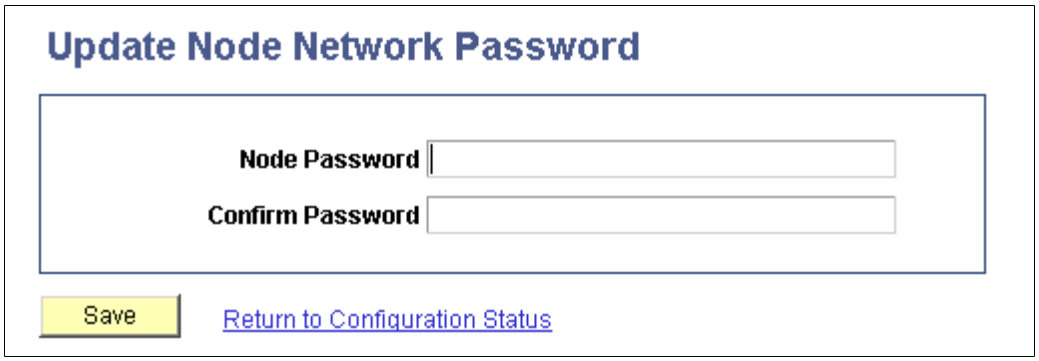
PeopleTools is delivered with an **IB_NETWORK** node. This node is used to perform functionality across all nodes in the integration network, including registering network nodes on participating systems.

The network participants determine the password to be used.

Important! This password must then be set on each system participating in the network. The password set must be identical on all systems participating in the integration network.

Use the Update Node Network Password page (IB_INTNET_NODEPWD) to enter the password for the network node. To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration** > **Configuration Status** and in the Additional Links section, click the **Node Network Password** link.

This example illustrates the Update Node Network Password page.



To set the network node password:

1. Access the Update Node Network Password page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration Status** and in the Additional Links section, click the **Node Network Password** link).
2. In the **Node Password** field, enter the network node password.
3. In the **Confirm Password** field, enter the network node password again.
4. Click the **Save** button.
5. Click the **Return to Configuration Status** link to return to the Configuration Status page.

Pinging Integration Network Nodes

This topic discusses how to:

- Use the Network Status page to ping integration network nodes.
- Ping network nodes.

Understanding Pinging Integration Network Nodes

Use the Ping Status button on the Network Status page (IB_VERIFYNETRS) to verify connectivity with remote nodes in the PeopleSoft network.

A successful ping indicates that you have correctly defined the selected remote PeopleSoft nodes in the integration network and that the local system can connect to them. A successful ping also indicates that the PeopleSoft integration partners represented by the remote nodes have correctly defined your node as part of their integration network and they should be able to perform a successful integration network ping to your system as well.

Prerequisites for Pinging Integration Network Nodes

Before you can ping local and remote integration network nodes:

- You and your PeopleSoft integration partners must have your Integration Broker systems configured and running.

  See Verifying and Managing Integration Gateway Configuration.

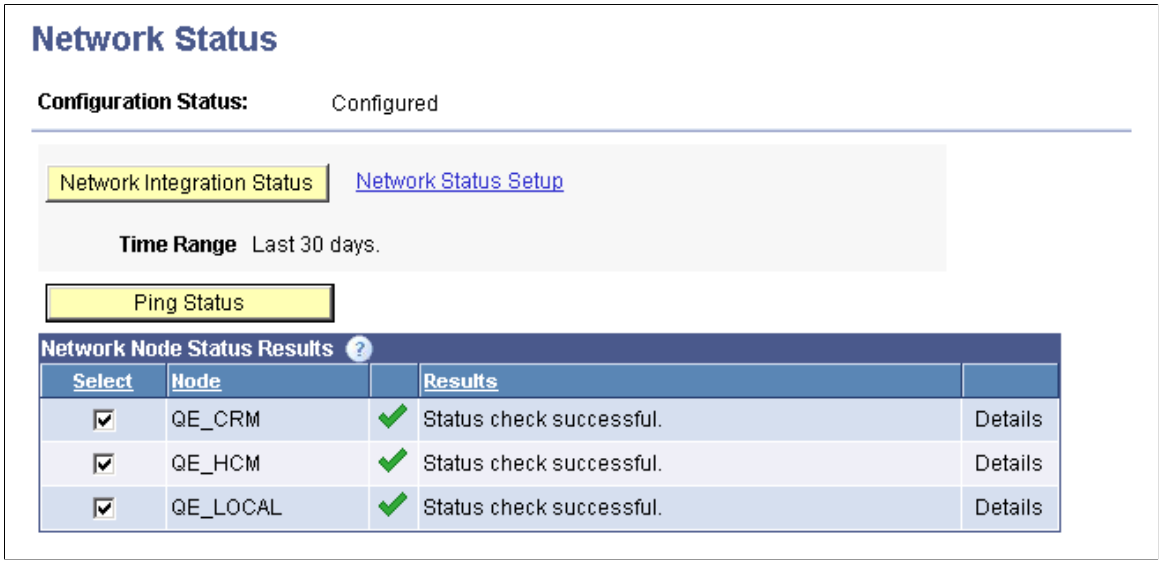
- You and your PeopleSoft integration partners must have the integration node network configured.

  See Viewing Node Network Status.

Using the Network Status Page to Ping Integration Network Nodes

To access the page Network Status page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Network Status.**

This example illustrates the fields and controls on the Network Status page. You can find definitions for the fields and controls used to ping network nodes later on this page.



When you access the page, all of the nodes defined in the integration network appear in the Network Node Status Results grid. You can select one or more nodes in the grid to perform the status check and then click the **Ping Status** button. The results of the ping appear in the Results column.

The Network Node Status Results grid in the previous example shows that the system successfully pinged all of the nodes defined in the integration network.

You can also use the Network Status page to verify that integrations are being processed between the default local node and remote network nodes. The following table lists and describes only the fields and controls on the page that are used in conjunction with pinging network nodes:

Field or Control	*Description*
Configuration Status	Displays the configuration status of the network. **Note:** The network must have a status of *Configured* to ping network nodes. The valid values are: • *Configured.* This value appears when the integration network is properly configured. • *Configuration Required.* This value appears as a link when the network is not configured, requires additional configuration, or is not properly configured. Click the link to access the Configuration Status page to identify and resolve any configuration issues. Verifying configuration status and using the Configuration Status page is described elsewhere. See Using the Configuration Status Page.
Ping Status	Use this page control to verify proper connectivity with remote nodes in the integration network.
Select	Check the box next to each node to include in ping status or integration processing checks. Clear the box next to each node to exclude in the checks.
Node	Displays the name of the node.
	The action was successful. The local node and remote PeopleSoft node successfully connect to each other in the integration network.
	The local node cannot connect to the remote PeopleSoft node.
Results	Area where a description of the results of the network status check and network ping appear.

Field or Control	*Description*
Details	If additional details of the network status and network ping results are available, a **Details** link appears. Click the link to view the additional information.

Pinging Network Nodes

To ping network nodes:

1. Access the Network Status page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Network Status**.)

   The nodes configured in the network appear in the Network Node Status Results grid.

2. Select the network nodes to ping.

   By default all defined network nodes are selected.

   Clear the **Select** box next to a node to exclude it from the ping. Check the box again to include it.

3. Click the **Ping Status** button.

The system displays the outcome of the ping in the Results column in the Network Node Status Results grid.

Deleting Nodes from the Integration Network

If you need to delete a node from the integration network there is a specific sequence of steps that you must follow.

If you simply remove the Jolt connection setting from the PeopleSoft Node Configuration page, the node will not appear as an in-network node in the Node Network page, but it will still appear in the Network Status page.

To delete a node from the node network you must follow these steps in the order presented:

1. On the Node Network page clear the **In-Network** box for the node to delete, and save the changes.
2. On the Gateways – PeopleSoft Node Configuration page delete the Jolt connection setting for the node. Save the changes and then click the **OK** button.

   The Gateways page appears.

3. Click the **Save** button.

If you delete the Jolt setting in the Gateways component without first clearing the **In-Network** box on the Node Network page, you must add the node back to the network and then remove it again, following the correct sequence. In this situation (where the Jolt setting for the node is deleted without first clearing the In- Network field for the node), the full procedure to properly delete the node is as follows:

1. On the Gateways – PeopleSoft Node Configuration page, add the Jolt connection setting for the node again. Save the changes and then click the **OK** button.

   The Gateways page appears.

2. Click the **Save** button.

3. On the Node Network page in the Integration Network WorkCenter, re-select the **In-Network** box for the node and save the changes.

4. On the Node Network page clear the **In-Network** box for the node and save the changes.

5. On the Gateways – PeopleSoft Node Configuration page delete the Jolt connection setting for the node. Save the changes and then click the **OK** button.

   The Gateways page appears.

6. Click the **Save** button.

Verifying Integration Processing in the Integration Network

This section discusses how to:

- Use the Network Status page.
- Use the Network Status Setup page.
- Use manual network integration processing verification.
- Use automated network integration processing verification.
- View network integration processing status results.

Understanding Network Integration Processing Verification

You can verify that integrations are being processed in the integration network. The system verifies that transactions between the local node and selected remote network nodes are not in error or time-out status. By the nature of the checks performed, the process also verifies connectivity with the selected remote network nodes.

You can manually perform integration processing verification on an as-needed basis for a day range that you define.

You can also set up automated integration processing verification for around-the-clock overage. In addition to setting the day range for processing verification, you also set options to define the interval at which processing verification occurs and define persons in your organization to notify should the system detect errors or failure in integration processing.

To perform manual or automated integration processing verification, use the Network Status page and the Network Status Setup page.

Note that manual and automated processing verification determines only if integrations are being processed between selected nodes. To monitor network transactions or track network transactions, use the Integration Network Monitor and the Integration Network Transactional Tracker, respectively.

Related Links

Using the Integration Network Monitor
Using the Integration Network Transactional Tracker

Understanding Integration Processing Diagnostics

When you verify integration processing between network nodes, whether it be manual or automated verification, you can specify that the system return diagnostics if any errors occur on the remote system. When you enable the diagnostic feature the system returns diagnostic information such as dispatcher name, number of active handlers, and so on. You can also choose to return information such as the number of transactions in retry status, memory available, and so on.

Prerequisites for Integration Processing Verification

Before you can perform manual or automated integration network processing verification with remote PeopleSoft nodes in the integration network:

- You and your PeopleSoft integration partners must have your Integration Broker systems configured and running.

  See Verifying and Managing Integration Gateway Configuration.

- You and your PeopleSoft integration partners must have the integration network node network configured.

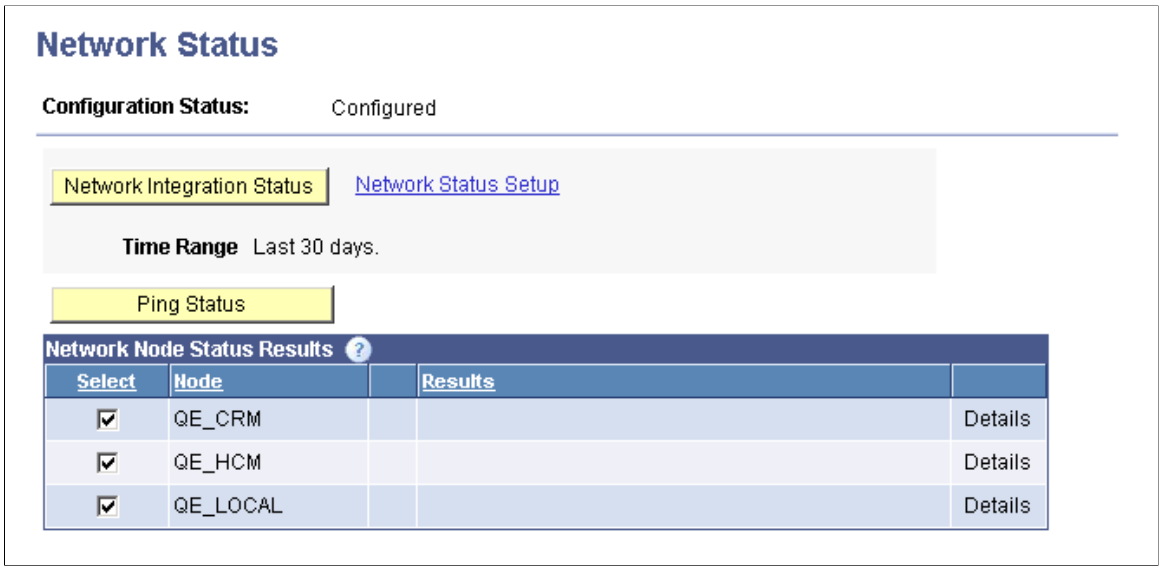
  See Viewing Node Network Status.

Using the Network Status Page

Use the Network Status page to manually verify that integrations are being processed between the local node and remote nodes defined in the integration network. Use the page to also access options to set up and use automated integration processing status verification.

To access the Network Status page (IB_VERIFYNETRS), select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Network Status.**

This example illustrates the fields and controls on the Network Status page. You can find definitions for the fields and controls later on this page.



The Network Status page features fields and controls used to access integration network configuration options, perform network integration status verification, and to ping integration network nodes. The following table lists and describes only the fields and controls on the page that are used in conjunction with verifying integration processing in the integration network:

Field or Control	***Description***
Configuration Status	Displays the configuration status of the network. **Note:** The network must have a status of *Configured* to verify integration processing among network nodes. The valid values are: • *Configured.* This value appears when the integration network is properly configured. • *Configuration Required.* This value appears as a link when the network is not configured, requires additional configuration, or is not properly configured. Click the link to access the Configuration Status page to identify and resolve any configuration issues. Verifying configuration status and using the Configuration Status page is described elsewhere. See Using the Configuration Status Page.

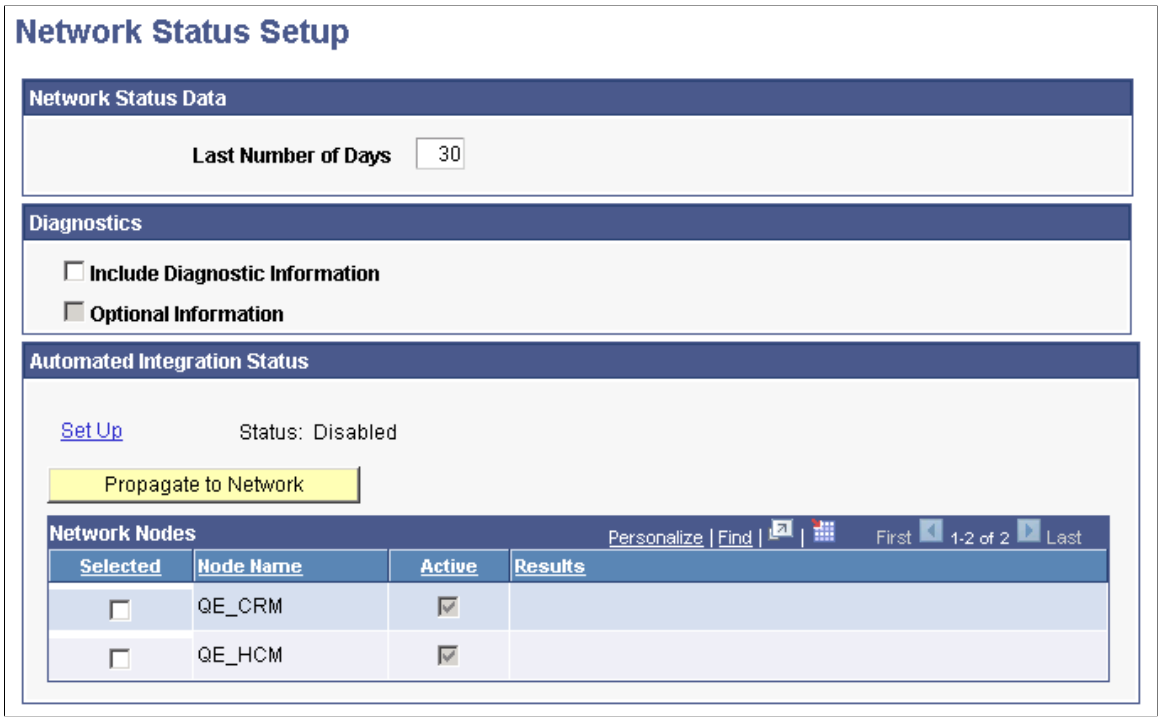
Field or Control	*Description*
Network Integration Status	Use this page control to manually verify the status of integrations with remote nodes defined in the integration network.
Network Status Setup	Click the link to access the Network Status Setup page and options to set the number of days for which to provide status, include diagnostics in status results, and set up automated integration status checking.
Time Range	Indicates the time range the system uses for reporting on the network status. Use the **Network Status Setup** link to modify this value.
Select	Check the box next to each node to include in ping status or integration processing checks. Clear the box next to each node to exclude in the checks.
Node	Displays the name of the network node.
Results	Area where a description of the results of the network status check and network ping appear. Viewing the results of this status check is described later in this topic.
Details	If additional details of the processing check are available, a **Details** link appears. Click the link to view the additional information. Viewing details of this status check is described later in this topic.

Using the Network Status Setup Page

Use the Network Status Setup page (IB_INTNET_STATSET) to set options when manually verifying integration processing between the default local node and remote PeopleSoft nodes, and to set up and use automated network integration processing verification.

To access the Network Status Setup page, from the Network Status page, click the **Network Status Setup** link.

This example illustrates the fields and controls on the Network Status Setup page. You can find definitions for the fields and controls later on this page.



The page features the following fields and controls:

Field or Control	***Description***
Last Number of Days	Enter the number of days for which to verify the processing of integrations between selected network nodes. The default value is *30* days.
Include Diagnostic Information	(Optional.) Select the box to generate system diagnostic information, such as dispatcher name, number of active handlers, and so on, should errors occur.
Optional Information	(Optional.) Select the box to generate information such as the number of transactions in retry status, memory available, and so on, in addition to system diagnostic information. This control is enabled only after the **Include Diagnostic Information** box is selected.
Set Up	Click the link to access the Automated Integration Status page to enable and set options for automated integration status monitoring.

Field or Control	*Description*
Disabled/Enabled	This read-only field displays the status of the automated integration status monitoring feature. By default the feature is *Disabled.*
Propagate to Network	Click the link to propagate the monitoring settings to selected nodes in the Network Nodes grid.
Selected	Select the box next to a network node to include it in automated integration processing verification.
Node Name	Indicates the name of a remote node defined in the integration network.
Active	A check in the box indicates the remote network node is active.
Results	When you propagate the verification setting to a network node, the results of the action appear in this field.
Return to Network Status	Click the link to return to the Network Status page.

Using Manual Network Integration Processing Verification

This section discusses how to:

- Set up manual network integration processing verification options.
- Manually verify network integration processing.
- View results of manual network integration processing verification.
- View integration processing status details.
- View diagnostic information.

Setting Up Manual Integration Verification Options

You can set options on the Network Status Setup page to specify the number of days for which to verify processing and enable the return of diagnostic information if the system detects errors in integration processing.

To set up manual integration status options:

1. Access the Network Status Setup page (**PeopleTools** > **Integration Broker** > **Integration Network** > **Administration** > **Network Status** and click the **Network Status Setup** link. The Network Status Setup page appears.
2. In the **Last Number of Days** field, enter the number of days for which to check processing status.

3. Select the **Include Diagnostic Information** box for the system to return diagnostics if the system determines integrations are not being processed.
4. Select the **Optional Information** box for the system to return additional diagnostics information if the system determines integrations are not being processed.
5. Click the **Save** button.
6. Click the **Return to Network Status** link to return to the Network Status page.

Manually Verifying Network Integration Status

To manually verify network integration status:

1. Access the Network Status Setup page (**PeopleTools** > **Integration Broker** > **Integration Network** > **Administration** > **Network Status.** The Network Status page appears.
2. In the Network Nodes Status Results section select the nodes to include in the verification.

   By default, all network nodes are selected. To exclude a node, check the **Select** box next to the node name. To include the node, check the **Select** box again.
3. Click the **Network Integration Status** button.

The results of the status verification appear in Network Nodes Status Results grid. Viewing the results is described elsewhere in this topic.

Viewing Results of Manual Network Integration Processing Verification

After you perform a manual network integration processing verification, the results appear on the Network Status page in the Network Node Status Results grid.

The following table lists the possible results from manual network integration processing verification:

Field or Control	*Description*
	A green check mark icon indicates that integrations are being processed between the default local node and the indicated remote network node.

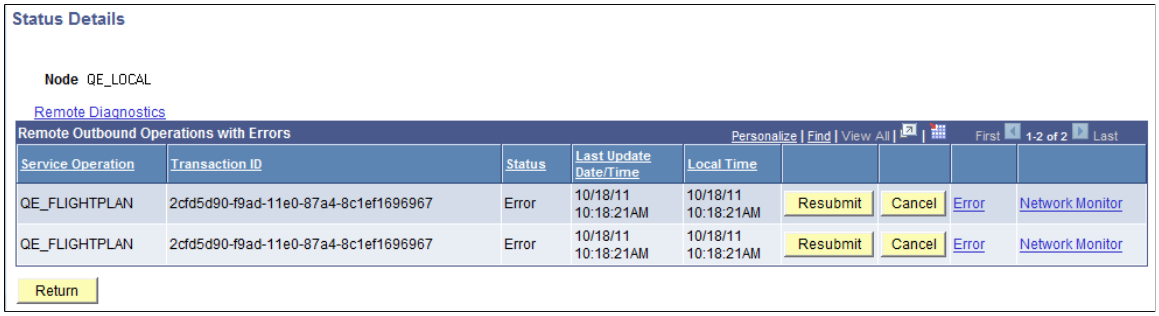
Field or Control	*Description*
	A yellow triangle icon indicates successful connectivity to the indicated remote node, however one of the following conditions exists: • The application server domain is not active. If the application server domain is not active, a **Network Monitor** link appears in the results grid. Click the link to access the Network Monitor and click the **Domain Status** link to activate the domain. • There are integrations on the local or remote node that are in *Timeout, Standby,* or *Error* status. If integrations are in any of these states, a **Details** link appears in the results grid. Click the link to access the Status Details page to obtain transaction ID, transaction date, and other details for troubleshooting the integrations in the Service Operations Monitor. In addition, if you selected to include integration processing diagnostics, can you access the diagnostic information form the Status Details page.
	A red “X” icon indicates that the integration status check failed. A **Details** link appears in the results grid. Click the link to view the Status Details page to view transaction errors by node. The system displays additional information per transaction to easily access information in the Network Monitor and Transactional Tracker. In addition, if you selected to include integration processing diagnostics, can you access the diagnostic information form the Status Details page.

View Integration Processing Status Details

Use the Status Details page to view additional information, if available, about the processing status between the default local node and a remote network node.

To access the Status Details page from the Network Node Status Results grid, click the **Details** link.

This example illustrates the fields and controls on the Status Details page. You can find definitions for the fields and controls later on this page.



The page features the following fields and controls:

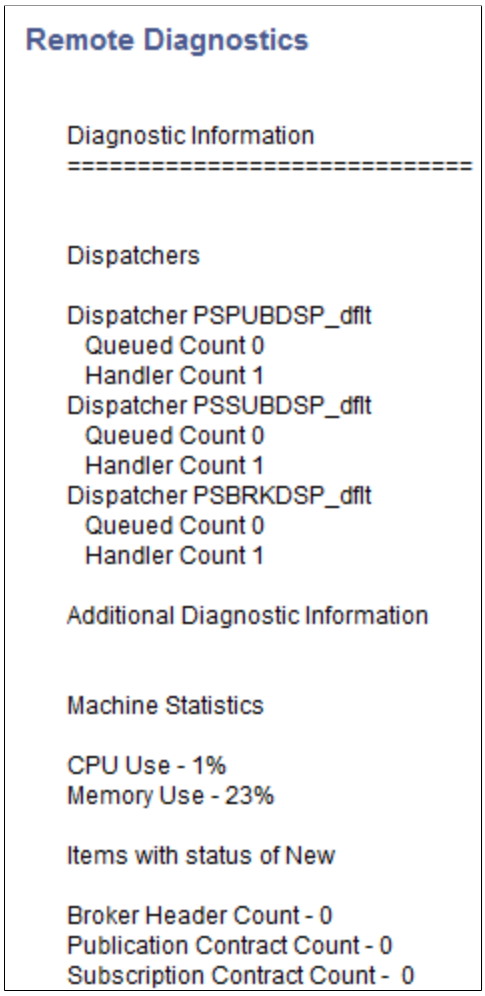
Field or Control	***Description***
Node	Name of the sending node.
Diagnostics	Click the link to view diagnostic information.
Inactive Domain	This link appears only if the pub/sub domain on the remote node is inactive. Click the link to activate the remote domain.
Remote Outbound Operations with Errors	This grid provides access to information and actions you can take on errors on outbound service operations to remote network nodes.
Remote Inbound Operations with Errors	This grid provides access to information and action you can take on errors on inbound service operations from remote network nodes.
Node	Name of the local default node.
Diagnostics	Click the link to view diagnostic information.
Service Operation	Service operation name.
Transaction ID	Unique numeric transaction identifier.

Field or Control	*Description*
Status	Indicates the status of the transaction. The possible values are the same that can appear in the Service Operations Monitor for asynchronous and synchronous transactions, including *Error,* **Timeout,** *Standby* and others. See “Understanding Asynchronous Service Operations Statuses” (Integration Broker Service Operations Monitor) and “Understanding Synchronous Service Operation Statuses” (Integration Broker Service Operations Monitor)
Last Update Date/Time	Indicates the update date/time the transaction was last updated. The time displayed is that which is defined on the remote system.
Local Time	Indicates the time as defined on the local system.
Resubmit	Click the button to resubmit the transaction for processing.
Cancel	Click the button to cancel the transaction for processing.
Error	Click the link to view additional error details, when available.
Network Monitor	Click the link to access the transaction in the Network Monitor for additional information and troubleshooting.
Return	Click the button to return to the Network Status page.

Viewing Diagnostic Information

If on the Network Setup Options page you selected the diagnostic option, the system will return diagnostic information when integration processing errors are detected on the remote system. The information appears in the Remote Diagnostics page. To access the page, click the **Diagnostics** link on the Status Details page.

This example illustrates the Remote Diagnostics page.



Using Automated Integration Processing Verification

This section discussed how to:

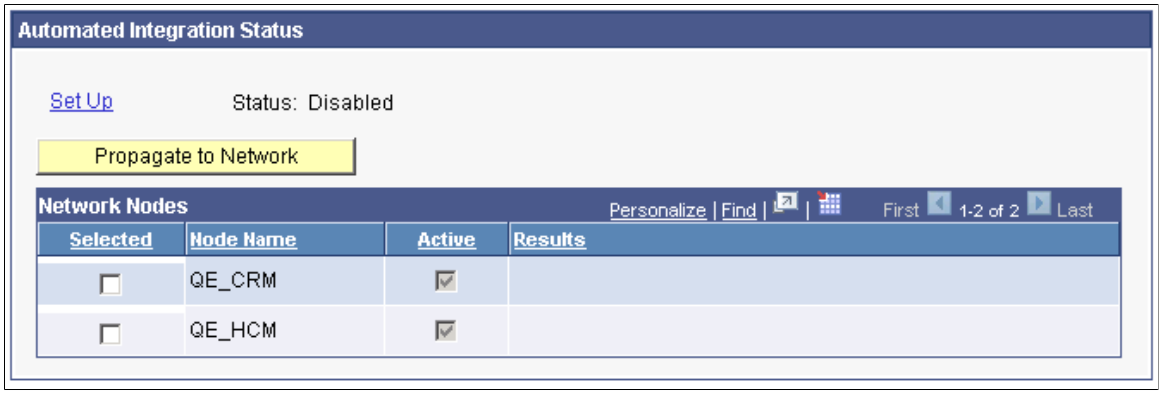
- View the status of automated network integration processing verification.
- Enable and set up automated network integration processing verification.
- Propagate integration processing verification to network nodes.
- View results of automated network integration processing verification.

Viewing the Status of Automated Integration Processing Verification

The Auto Network Sync section of the Network Status Setup page provides a read-only **Enabled**/**Disabled** toggle field that indicates if automated integration processing verification is enabled.

To access the Network Status Setup page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Network Status.** The Network Status page appears. Click the **Network Status Setup** link.

This example illustrates the fields and controls in the Auto Network Sync section of the Network Status Setup page.

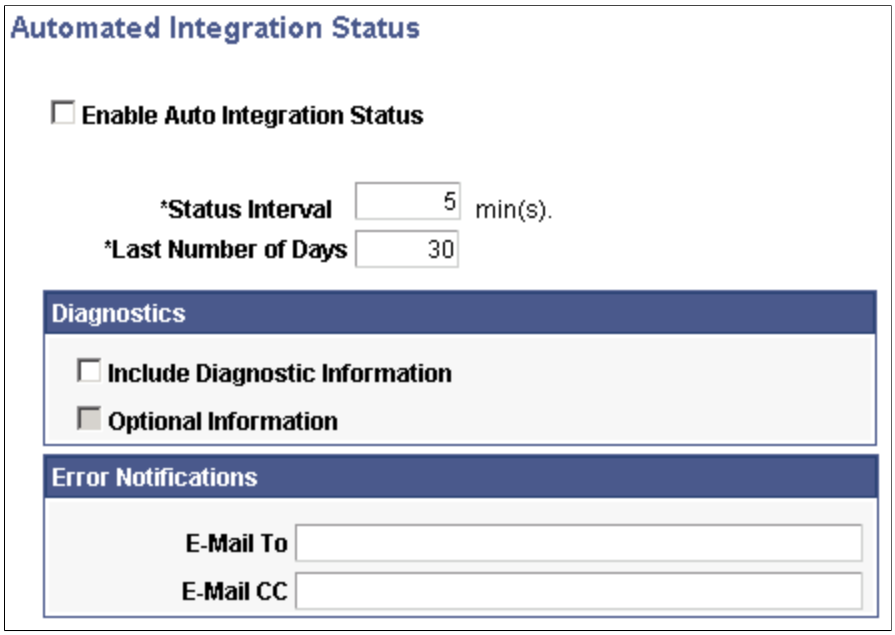


In the previous example, the term *Disabled* appears in the Auto Network Sync section of the page, indicating that automated integration processing verification is not enabled.

Enabling and Setting Up Automated Integration Processing Verification

Use the Automated Integration Status page (IB_INTNET_STAT_SEC) to enable and set up automated integration processing verification. To access the page, from the Network Status Setup page click the **Set Up** link in the Auto Network Sync section of the page.

This example illustrates the fields and controls on the Automated Integration Status page. You can find definitions for the fields and controls later on this page.



The page features the following fields and controls:

Field or Control	*Description*
Enable Auto Integration Status	Select the box to enable automated integration processing verification.
Status Interval	Enter the frequency in minutes that the system performs integration processing verification. The default value is *5* minutes.
Last Number of Days	Indicates the time range the system uses for reporting integration processing verification. The default value is *30* days.
Include Diagnostic Information	(Optional.) Select the box to generate system diagnostic information, such as dispatcher name, number of active handlers, and so on, should an error occur.
Optional Information	(Optional.) Select the box to generate information such as the number of transactions in retry status, memory available, and so on, in addition to system diagnostic information. This control is enabled only after the **Include Diagnostic Information** box is selected.
E-Mail To	Enter the email address to notify in the event of an error. Separate multiple addresses with a semicolon.
E-Mail CC	(Optional.) Enter the email address to copy on a notification if an error occurs. Separate multiple address with a semicolon.

To enable and set up automated integration processing verification:

1. Access the Automated Integration Status page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Network Status.** The Network Status page appears. Click the **Network Status Setup** link. The Network Status Setup page appears. Click the **Set Up** link. The Auto Network Sync page appears.
2. Select the **Enable Auto Integration Status** box.
3. In the **Status Interval** field enter interval between checks, in minutes, for verifying processing.
4. In the **Last Number of Days** field enter the number of days for which to verify processing.
5. (Optional.) Select the **Include Diagnostic Information** box for the system to return diagnostics if the system determines integrations are not being processed.

6. (Optional.) Select the **Optional Information** box for the system to return additional diagnostics information if the system determines integrations are not being processed.
7. In the **E-Mail To** field, enter one or more email addresses to notify if the system detects errors in integration processing.
8. In the **E-Mail CC** field, enter one or more email addresses to copy on notifications of errors in integration processing.
9. Click the **OK** button.

Propagating Network Integration Processing Verification to Network Nodes

After you enable and select automated integration processing verification options, use the Network Status Setup page to select the nodes with which to perform network integration processing verification and to propagate the verification options to those nodes.

The propagate to network feature enables you to enable automated network integration processing verification on selected network nodes and push verification options to those nodes, without the need to manually set the options on each system.

To propagate network integration processing verification to network nodes:

1. Access the Network Status Setup page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Network Status.** The Network Status page appears. Click the **Network Status Setup** link. The Network Status Setup page appears).
2. In the Network Nodes grid in the Automated Integration Status section of the page, check the box next to each node to include in integration processing verification.
3. Click the **Propagate to Network** button.

Results of the action appear in the Network Node Status Results grid.

Viewing Results of Automated Network Integration Processing Verification

When automated network integration processing verification is enabled and the system detects processing errors, it sends a notification to the email address(es) defined in the Auto Network Sync page. If during set up you selected the diagnostics option, diagnostic information is included in the notification.

Upon receipt of a notification, troubleshoot any errors by using pages and tools provided in the Integration Network, including:

- Manual network processing verification described previously in this topic.
- Integration Network Monitor. See Using the Integration Network Monitor
- Transactional Tracker. See Using the Integration Network Transactional Tracker

Introspecting and Deploying Network Integrations

This section discusses how to:

- Use the Search for Items-Introspection/Deployment page.
- Search for items to introspect and deploy.
- Select integrations to introspect and deploy.
- Select nodes for introspection and deployment.
- Verify nodes and services to introspect.

Understanding Introspecting and Deploying Integrations

The Integration Network features a series of pages that enable you to introspect and deploy services on remote nodes.

The system can introspect and deploy integrations on any node defined in the integration network. However, to introspect integrations on the local node, a handler and a local routing must exist on any integration that you introspect.

Important! To introspect a service operation on the local node, a handler and a local routing must be defined for the service operation.

Introspection Checks

During the introspection process, the system checks the following items for the selected service operations for the selected nodes:

- Service operations are active.
- Routings.
    - Routings exist.

      Note that routings are checked from the node where introspection was invoked.

      Physical routings, deployment configuration routings and default routings are all potential routings that can be auto-generated based on versions available. The prioritization for routing checks is 1.) physical routing, 2.) deployment configuration routing, and finally 3.) default routing.
    - Alias names match.

      The alias name match is verified on the physical routing, the deployment configuration, or the default routing.

      The system derives the alias from the name and version of the service operation.
    - Sending and receiving nodes match.
    - No multiple target aliases exist. (Synchronous service operations only.)

Introspection Checks (REST)

During the introspection process, the system checks the following items for the selected REST service operations for the selected nodes:

REST as consumer (source):

- Service operation is active.
- Routing exists.

  Check for an any-to-local routing on the target with an alias name that is the same as alias name on source (local-to-WADL) defined as the REST provider on the target. If no such routing exists, the system check that the service operation.version is defined as a REST provider on the target.
- Handler exists.

  The system checks that an OnRequest handler exists on the target system for the any-to-local routing.
- REST base URL update.

  The system updates the REST base URL on the source system to <REST target Location endpoint>/< External Alias from Any to Local routing> on the target system.
- Service operation permissions exist.

REST as provider (target):

- Service operation is active.
- Routing exists.

  Check for a local-to-WADL routing on the target with an alias name the same as alias name on source (any-to-local) defined as REST consumer on target. If no such routing exists, the system checks that the service operation.version is defined as a REST consumer on the target.
- Handler exists.

  The system checks that an OnRequest handler exists on the source system for the any-to-local routing
- REST base URL update.

  The system updates the REST base URL on the target system to <REST target Location endpoint>/< External Alias from Any to Local routing> from the source system.
- Service operation permissions exist.

Deployment Processing (Non-REST and REST)

In many cases during deployment processing, the system can activate service operations, handlers, and routings that are found to be inactive during the introspection process. The system can also frequently create routings if none are found to exist during introspection. However, there are some situations where manual intervention is required to resolve routing issues and handler issues. And in all cases where service operation permissions do not exist, manual intervention is required to assign permissions.

Related Links

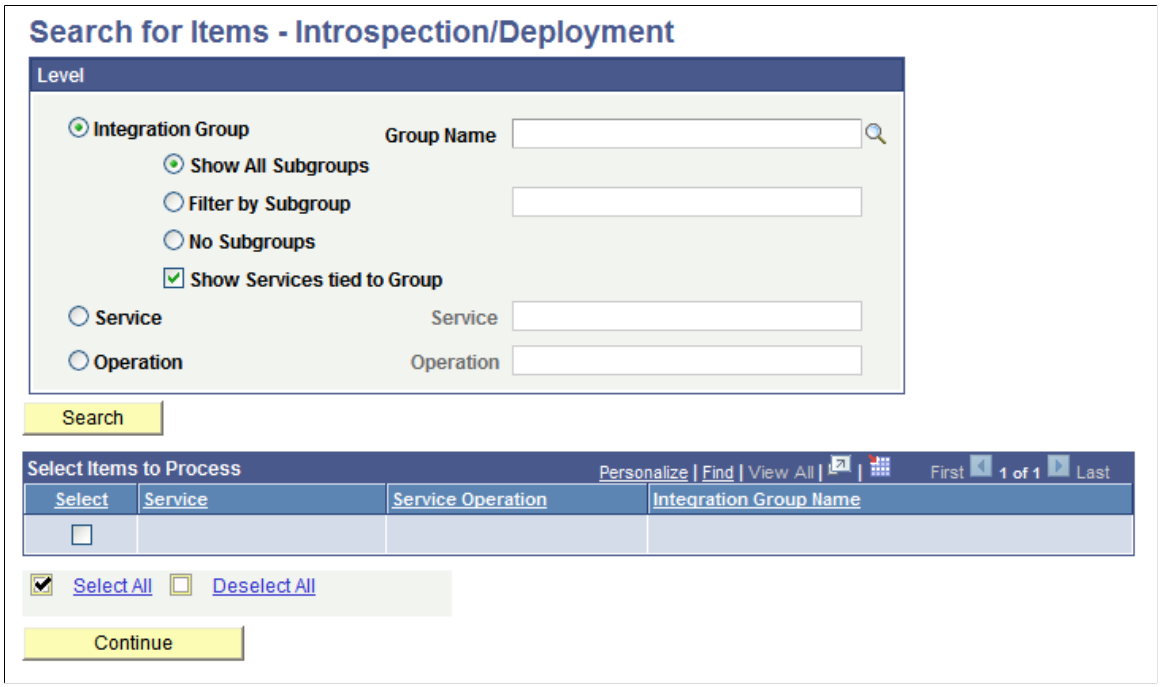
"Configuring Routing Definitions for Deployment" (Integration Broker)

Using the Search for Items–Introspection/Deployment Page

The Integration Network features a Search for Items – Introspection/Deployment page (IB_INTNETWORK2) that you use to search for select integrations to introspect and deploy.

To access the Search for Items – Introspection/Deployment page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Introspect/Deploy Integrations.**

This example illustrates the fields and controls on the Search for Items–Introspection/Deployment page . You can find definitions for the fields and controls later on this page.



Searching for Integrations to Introspect and Deploy

You can search for integrations to introspect and deploy at the integration group, service, or service operation level.

Use one of the following options to search for an integration:

Field or Control	*Description*
Integration Group	Select the radio button to search for integrations to introspect and deploy by integration group. Use one of the following integration group filters: • *Show All Subgroups.* (Default.) Select the radio button to display all integration subgroups associated with integration groups in the system. • *Filter by Subgroup.* Select the radio button to search for a specific integration subgroup and enter its name in the field. • *No Subgroups.* Select the radio button to omit integration subgroups from your search. • *Show Services Tied to Group.* Check the box to include the services tied to an integration group in the search results. By default this option is selected.
Group Name	If searching by integration group, enter the integration group name to search, or click the **Lookup** button to search for one.
Service	Select the radio button to search for integrations to introspect and deploy by service. When you select this option a **Lookup** button appears next to the Service field. Click the **Lookup** button to search for a service to introspect and deploy or enter the name directly in the field.
Operation	Select the radio button to search for integrations to introspect and deploy by service operation. When you select this option a **Lookup** button appears next to the **Operation** field. Click the **Lookup** button to search for a service operation to introspect and deploy or enter the name directly in the field.

After selecting one of the search options, click the **Search** button at the bottom of the page.

Results of the search appear in the Select Items to Process grid at the bottom of the page.

Selecting Integrations to Introspect and Deploy

When you search for integrations to introspect and deploy to remote PeopleSoft nodes in the integration network, the results appear in the Select Items to Process grid at the bottom of the Search for Items – Introspection/Deployment page. Use the grid to select the integrations to introspect and deploy.

The following controls and fields appear in the Select Items to Process grid:

Field or Control	*Description*
Select	Check the box to introspect and deploy the integration. Check the box again to clear the selection.
Service	Name of service in the integration.
Service Operation	Name of the service operation in the integration.
Integration Group	Name of the integration group to which the integration belongs.

By default, all integrations returned in the search are selected for introspection and deployment.

To select an integration to introspect and deploy:

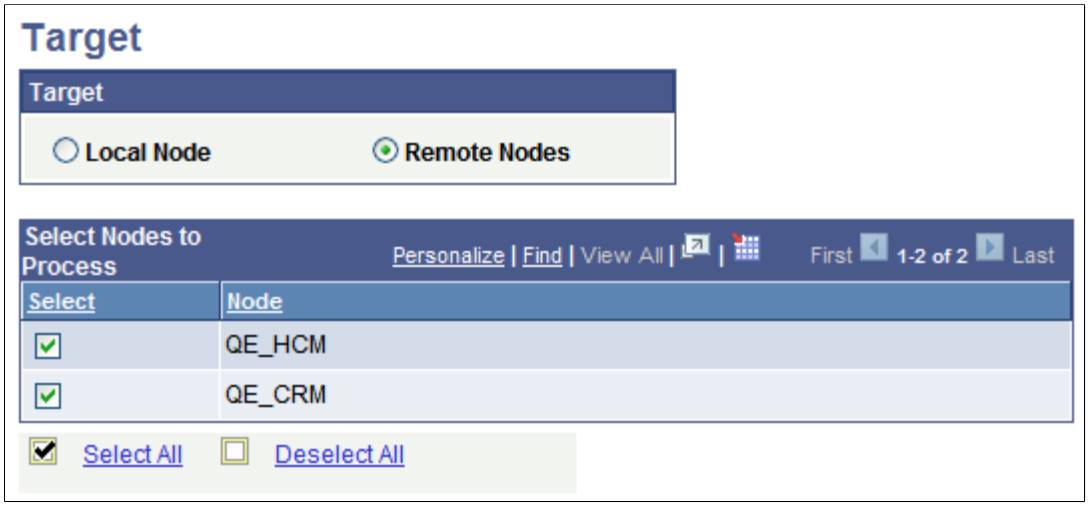
1. Use one of the following methods to select integrations from the grid to introspect and deploy:
    - Check the **Select** box next to individual integrations to select them or deselect them.
    - Click the **Select All** link to select all the integrations in the results grid.
    - Click **Deselect All** link to deselect all the integrations in the results grid.
2. Click the **Continue** button to select access the Target page and to select the remote PeopleSoft nodes to introspect and deploy.

Selecting Nodes for Integration Introspection and Deployment

Use the Target page (IB_INTNETTARGET) to select nodes for integration introspect and deploy integrations. You can select the default local node, for local-to-local integrations, or select one or more remote nodes.

To access the Target page, from the Search for Items – Introspection/Deployment page, click the **Continue** button.

This example illustrates the Target page.



To select nodes for introspection and deployment:

1. Access the Target page. (From the Search for Items – Introspection/Deployment page, click the **Continue** button.)
2. Perform one of the following actions:
   - Select the **Local Node** radio button for local-to-local integrations.
   - Select the **Remote Nodes** radio button to introspect and deploy integrations to remote PeopleSoft nodes.

Selecting the Local Default Node for Integration Introspection and Deployment

To select the local default node for integration introspection and deployment:

1. Access the Target page. (From the Search for Items – Introspection/Deployment page, click the **Continue** button.)
2. Select the **Local Node** radio button.
3. Click the **Continue** button.

Selecting Remote PeopleSoft Nodes for Integration Introspection and Deployment

To select remote PeopleSoft nodes for integration introspection and deployment:

1. Access the Target page. (From the Search for Items – Introspection/Deployment page, click the **Continue** button.)
2. Select the **Remote Nodes** radio button.
3. Use one of the following methods to select nodes for integration introspection and deployment:

   By default, all remote nodes in the integration network are selected for introspection and deployment.

   - Check the **Select** box next to individual integrations to select them or deselect them.

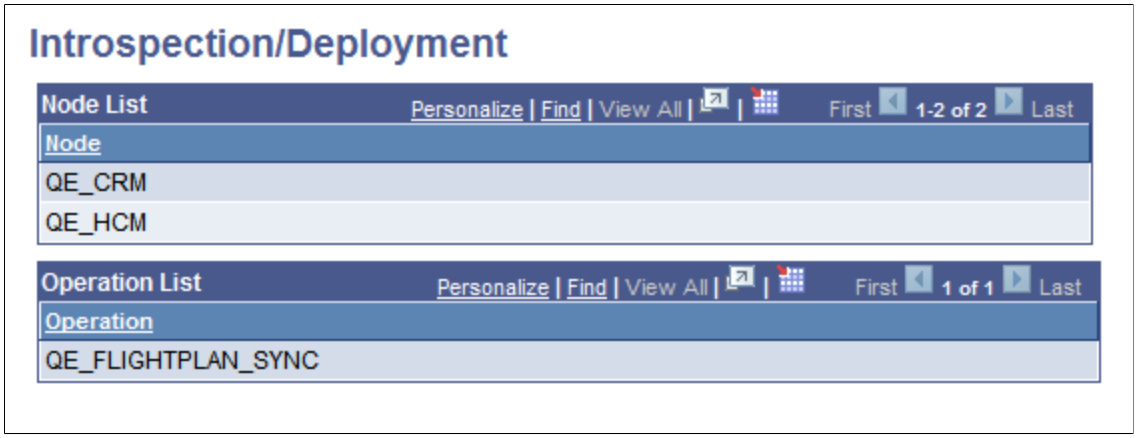
- Click the **Select All** link to select all the integrations in the results grid.
- Click **Deselect All** link to deselect all the integrations in the results grid.

4. Click the **Continue** button to preview the integrations to be introspected and deployed.

Verifying Nodes and Services to Introspect

The Introspection/Deployment page (IB_INTNETWORK4) displays the nodes and service operations you selected for integration and deployment in the previous step.

This example illustrates the Introspection/Deployment page. The example shows the nodes and service operations selected for introspection and deployment.



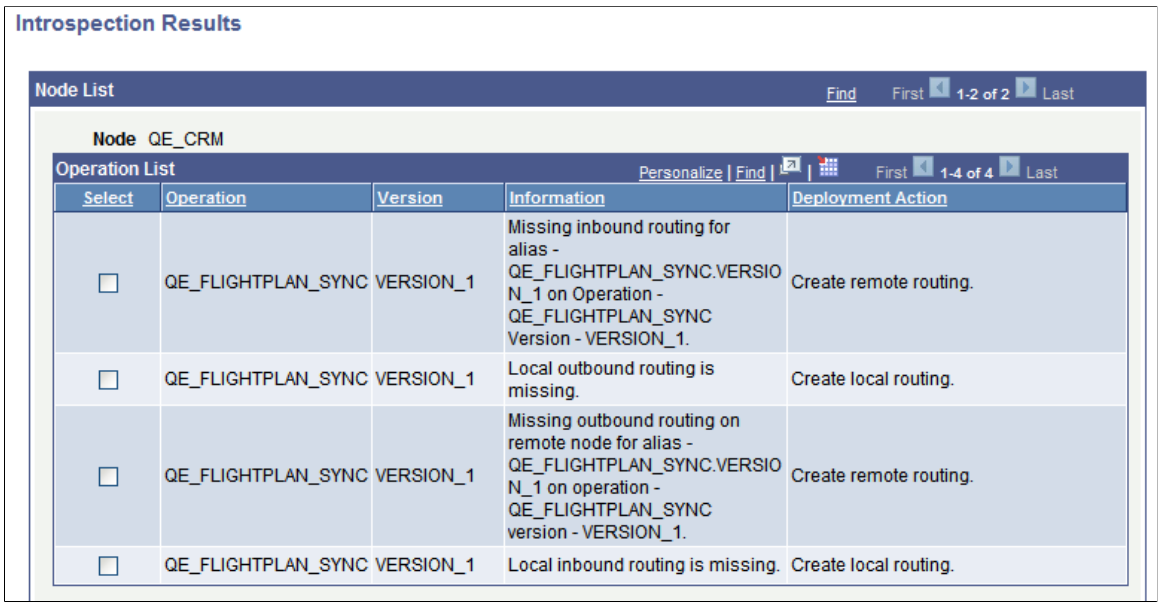
To make changes, use the **Previous Step** button located at the bottom of the page to return to the previous pages and make any necessary changes.

If the information is satisfactory, click the **Introspect** button to introspect the nodes in the node list. The results appear in the Introspection Results page discussed in the next section.

Viewing Introspection Results and Deploying Actions

The results of the introspection appear on the Introspection Results page (IB_INTNETWORK3_SEC).

This example illustrates the Introspection Results page.



Introspection Results

Node List — Find — First 1-2 of 2 Last

Node QE_CRM

Operation List — Personalize | Find | First 1-4 of 4 Last

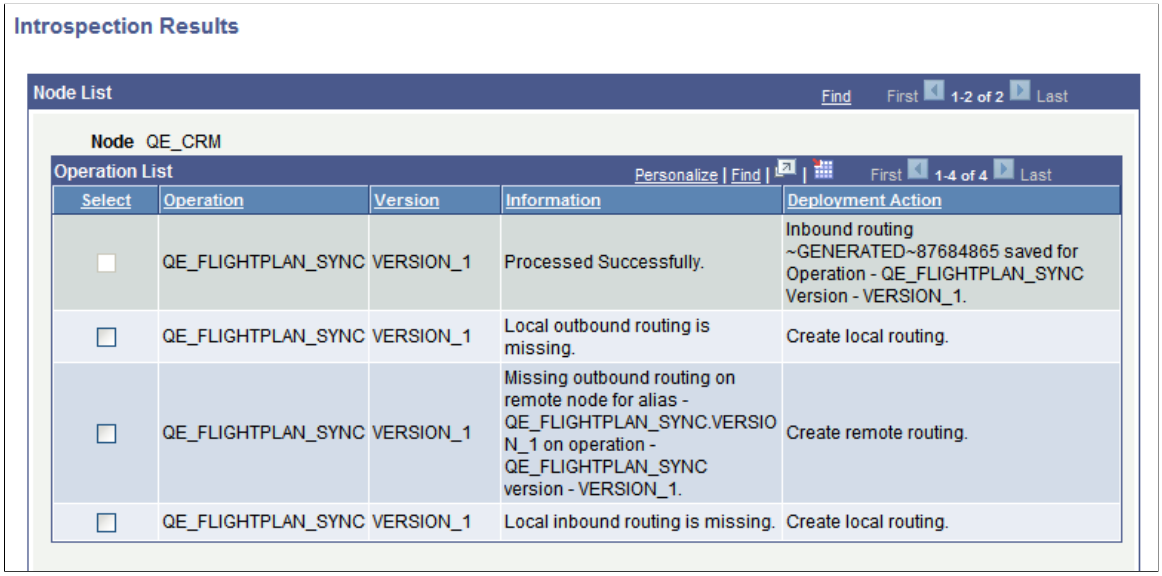
Select	Operation	Version	Information	Deployment Action
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Missing inbound routing for alias - QE_FLIGHTPLAN_SYNC.VERSION_1 on Operation - QE_FLIGHTPLAN_SYNC Version - VERSION_1.	Create remote routing.
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Local outbound routing is missing.	Create local routing.
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Missing outbound routing on remote node for alias - QE_FLIGHTPLAN_SYNC.VERSION_1 on operation - QE_FLIGHTPLAN_SYNC version - VERSION_1.	Create remote routing.
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Local inbound routing is missing.	Create local routing.

The Introspection Results page displays results for all nodes introspected, however due to space constraints the previous example shows only the results for the *QE_CRM* node.

The Introspection Results page lists any needed deployment actions that you can correct using the page. To deploy an action, select a row in the grid and click the **Deploy** button.

For example, in the previous example, the first row in the Operation List grid for the **QE_CRM** grid shows that an inbound routing does not exist for the routing alias *QE_FLIGHTPLAN_SYNC.VERSION_1*. To create the routing, check the Select box for the item and click the **Deploy** button. The system generates a routing.

This example illustrates the Introspection Results page after the system performs the deployment action. The example shows the page after creating an inbound routing for the routing alias *QE_FLIGHTPLAN_SYNC.VERSION_1*



Introspection Results

Node List — Find — First 1-2 of 2 Last

Node QE_CRM

Operation List — Personalize | Find | First 1-4 of 4 Last

Select	Operation	Version	Information	Deployment Action
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Processed Successfully.	Inbound routing ~GENERATED~87684865 saved for Operation - QE_FLIGHTPLAN_SYNC Version - VERSION_1.
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Local outbound routing is missing.	Create local routing.
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Missing outbound routing on remote node for alias - QE_FLIGHTPLAN_SYNC.VERSION_1 on operation - QE_FLIGHTPLAN_SYNC version - VERSION_1.	Create remote routing.
☐	QE_FLIGHTPLAN_SYNC	VERSION_1	Local inbound routing is missing.	Create local routing.

In this example the system has created an inbound routing for the *QE_FLIGHTPLAN_SYNC.VERSION_1* service operation.

The status of the action appears in the Information field and the deployment action performed appears in the **Deployment Action** field.

Click the **Return to Target Page** link to return to the Target page select additional or different nodes to introspect. Click the **Return to Search** link to return to the Search for Items – Introspection/Deployment page to work with different integration groups, services, and service operations.

Setting Service Operation Permissions

This section discusses how to:

- Use the Service Operation Permissions page.
- Use the Web Service Access page.
- Use the Bulk Service Operation Permissions page.
- Search for service operations to set permissions.
- Set permissions for individual service operations.
- Set permissions for service operations in bulk.

Understanding Setting Service Operation Permissions

The PeopleSoft Integration Network features a Service Operations Permissions page (IB_HOME_PAGE10) that enables you to set and modify permissions for individual network services operations.

A Bulk Service Operation Permissions page (IB_HOME_PAGE10A) is provided for setting permissions for network service operations.

You can search for service operations for which to set permissions at the integration group, service, or service operation level.

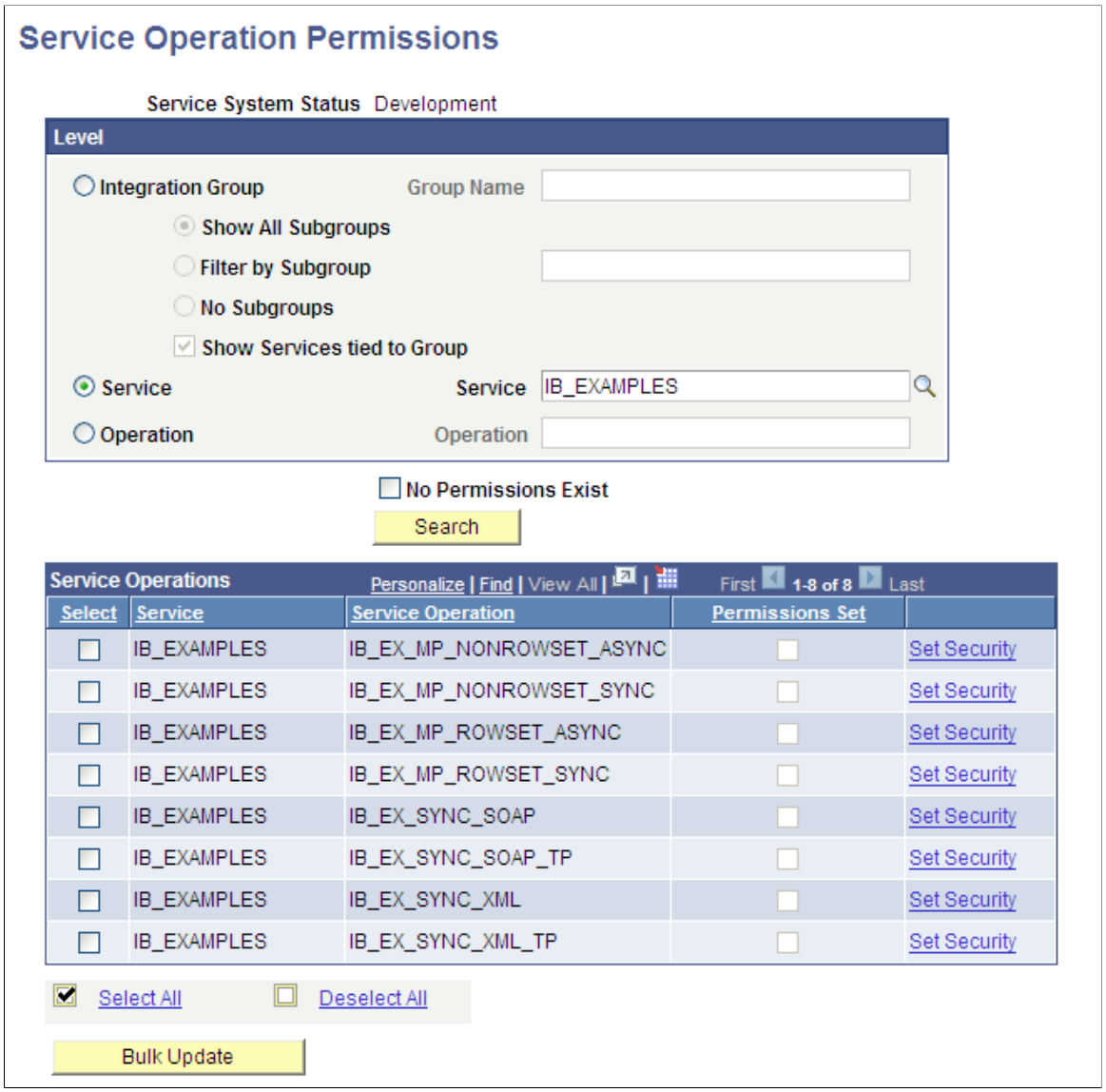
Note: You can set and modify service operation permissions on the local node only.

Using the Service Operation Permissions Page

Use the Service Operation Permissions page (IB_HOME_PAGE10) to set permissions for individual service operations.

To access the page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Service Operation Permissions.**

This example illustrates the fields and controls on the Service Operation Permissions page. You can find definitions for the fields and controls later on this page.



Use the options in the Level section to search for service operations for which to set permissions. The results of your search appear in the Service Operations grid at the bottom of the page.

The following fields and controls appear on the page:

Field or Control	***Description***
Service System Status	This field displays the service system status set on the database.

Field or Control	*Description*
Integration Group	Select the radio button to search for service operations by integration group. Use one of the following integration group filters: • Show All Subgroups. (Default.) Select the radio button to display all integration subgroups associated with integration groups in the system. • Filter by Subgroup. Select the radio button to search for a specific integration subgroup and enter its name in the field. • No Subgroups. Select the radio button to omit integration subgroups from your search. • Show Services Tied to Group. Check the box to include the services tied to an integration group in the search results. By default this option is selected.
Group Name	If searching by integration group, enter the integration group name to search, or click the Lookup button to search for one.
Service	Select the radio button to search for service operations by service. When you select this option a **Lookup** button appears next to the Service field. Click the **Lookup** button to search for a service to introspect and deploy or enter the name directly in the field.
Operation	Select the radio button to search for service operations by service operation. When you select this option a **Lookup** button appears next to the Operation field. Click the **Lookup** button to search for a service operation to introspect and deploy or enter the name directly in the field.
No Permissions Exist	Check the box to filter your search results for only those service operations for which no permissions exist.
Search	Click the button to search the database for service operations based on the search criteria entered.

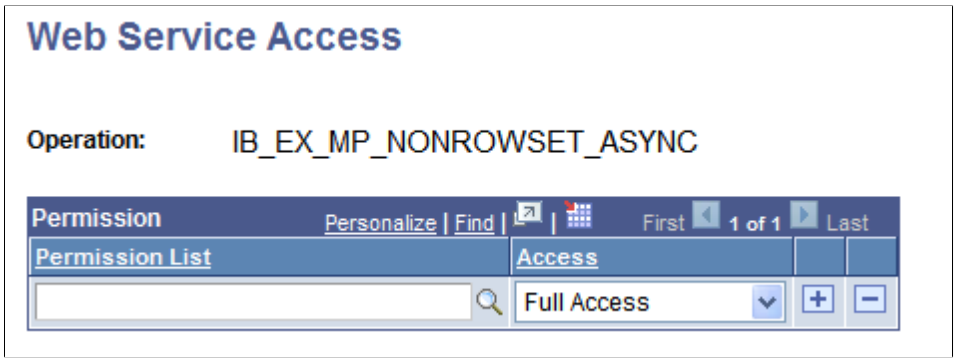
Field or Control	*Description*
Service Operations (grid)	This grid displays the search results and contains the following fields and controls: • Select. Select the box to include a service operation for bulk changes. • Service. Name of service. • Service Operation. Name of service operation. • Permission Set. This read-only field indicates if permissions have been set for a service operation. • Set Security. Click the link to access the Web Service Access page to set permissions for the selected service operation.
Select	Select the box to include a service operation for bulk changes.
Select All	Select the box to select all service operations for bulk changes to service operation permissions.
Deselect All	Select the box to deselect any service operations selected.
Bulk Update	Click the button to apply permissions in bulk to selected service operations.

Using the Web Service Access Page

Use the Web Service Access page (WS_ACCESS_IB) to define permissions for a service operation, including assigning a permission list and setting the access level to the service operation.

You access this page from the Integration Network from the Service Operation Permissions page. To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Service Operation Permissions.** After you select one or more service operations with which to work, in the Service Operation results grid at the bottom of the page, click the **Set Security** link.

This example illustrates the fields and controls on the Web Service Access page. You can find definitions for the fields and controls later on this page.



The page features the following fields and controls:

Field or Control	***Description***
Operation	Name of the service operation for which to set permissions.
Permission (grid)	This grid features the following fields and controls: • Permission List.Click the **Lookup** button to select a permission list to assign. • Access. Select an access level from the list.

Complete documentation for the Web Service Access page can be found elsewhere in the product documentation.

See “Defining Permissions” (Security Administration)

Using the Bulk Service Operation Permissions Page

Use the Bulk Service Operation Permissions page (IB_HOME_PAGE10A) to apply permissions to service operations in bulk.

You access this page from the Service Operation Permissions page. To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Service Operation Permissions.** After you select the service operations with which to work, click the **Bulk Update** button at the bottom of the page.

This example illustrates the fields and controls on the Bulk Service Operation Permissions page. You can find definitions for the fields and controls later on this page.

Any permissions that already exist for the service operations appear in the Bulk Permissions grid.

This page features the following fields and controls:

Field or Control	***Description***
Operation List (grid)	The grid displays the service operations to which to apply permissions. The grid features the following fields and controls: • Service. Name of the service. • Service Operation. Name of the service operation.
Bulk Permissions (grid)	This grid features the following fields and controls: • **Permission List.** Click the **Lookup** button to select a permission list to assign. • **Access.** Select an access level from the list.
Clear All	Click the link to clear all defined permission lists and access levels assigned to the service operations in the Operation List grid.

Field or Control	*Description*
Update	Click the button to apply the permissions selected in the Bulk Permissions grid to the service operations.
Return to Service Operation Permissions	Click the link to return to the Service Operation Permissions page.

Searching for Service Operations to Set Permissions

The information in this section applies to searching for service operations for applying permissions individually or in bulk.

To search for service operations to set permissions:

1. Access the Service Operation Permissions page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Service Operation Permissions.**
2. In the Level section of the page, select the options and enter the criteria to use to search the database for service operations.

   The fields and controls of the page are described previously in this topic.
3. Click the **Search** button.

Results of the search appear in the Service Operations grid at the bottom of the page.

You can now apply permissions individually to service operations or apply them in bulk. The next sections describe these procedures.

Setting Permissions for Individual Service Operations

The information in this section describes setting permissions for individual service operations by using the Web Service Access page. This section describes accessing the page using the Integration Network. Alternate methods for access this page are described elsewhere in the product documentation.

To set permissions for individual service operations:

1. Access the Web Service Access page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Service Operation Permissions.** Search for and select the service operation with which to work and click the **Set Security** link in the Service Operations grid)
2. From the **Permission List** drop-down list, select a permission list to assign to the service operation.
3. From the **Access** drop-down list, select an access level for the permission list.
4. (Optional.) Click the **Add Row** icon (+) to add additional permission lists and access levels to the service operation.
5. Click the **Save** button.

Related Links

"Defining Permissions" (Security Administration)

Setting Permissions for Service Operations in Bulk

To set permissions for service operations in bulk:

1. Access the Bulk Service Operations page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Administration** > **Service Operation Permissions.** Search for and select the service operations with which to work and click the **Bulk Update** button at the bottom of the page).
2. From the **Permission List** drop-down list, select a permission list to assign to the service operations in the Operation List grid.
3. From the **Access** drop-down list, select an access level for the permission list.
4. (Optional.) Click the **Add Row** icon (+) to add additional permission lists and access levels to the service operations.
5. Click the **Update** button.
6. Click the **Return to Service Operation Permissions** link to return to the Service Operation Permissions page.

Using the Integration Network Monitor

This section discusses how to:

- Set up the Network Monitor.
- Select nodes to monitor.
- Fetch and synchronize Network Monitor data.
- Monitor asynchronous network service operations.
- Monitor synchronous network service operations.
- Activate and deactivate network domains.
- Resubmit and cancel network transactions.

Understanding Using the Integration Network Monitor

System administrators use the Integration Network Monitor to monitor integrations among other PeopleSoft systems defined in the integration network. Many of the pages and functions of the Network Monitor resemble those in the monitoring pages of the Service Operations Monitor. The major differences are that with the Network Monitor you are not limited to monitoring activity on the local node; you can monitor activity any remote PeopleSoft node defined in the network.

Network Monitor Features

The Network Monitor provides the following features:

- Status on queues, nodes, and individual service operations.
- Control and administer domains that have publication and subscription (pub/sub) servers running against the current database.

  You can activate or deactivate domains, recover from stalls, and so forth.

Continue to use the Service Operation Monitor to perform the following tasks:

- View and edit XML.
- Archive data.
- Run batch processes to receive notification of issues affecting the messaging system.
- Administrative tasks, such as administer domains, view and maintain queue status, delete orphaned data after segment batch processing errors, define custom components for monitoring activities, and so on.

See the product documentation for Integration Broker Service Operations Monitor

Network Monitor Pages

The Network Monitor features the following pages:

Term	***Definition***
Monitor Overview	Use the Monitor Overview page (IB_INTNET_PUBCON) for a high-level overview of the status of asynchronous service operation transactions. You can group transactions by queue or service operation for viewing.
Operation Instances	The Operation Instances page (IB_INTNET_PUBHDR) enables you to monitor the status and details related to individual asynchronous service operation instances.
Publication Contracts	The Publication Contracts page (IB_INTNET_PUBCON) shows outbound publication transactions to send to remote nodes.
Subscription Contracts	The Subscription Contracts page (IB_INTNET_SUBCON) enables you to view transactions to which the local node subscribes. Subscription contracts for remote nodes do not appear.

Term	*Definition*
Synchronous Transactions	The Synchronous Transaction page (IB_INTNET_SYNC) enables you to view synchronous transactions.
Domain Status	The Domain Status page (IB_INTNET_DOMSTAT) enables you to view and activate domains in the integration network.

In addition, every page of the Network Monitor features these links:

Term	*Definition*
Monitor Setup	Click the link to access the Network Monitor Setup page to modify setup options, such as the number of rows to return for a query.
Search Details	Click the link to retrieve a summary detail for a specific transaction. When you enter a transaction ID for an asynchronous transaction the Asynchronous Details page appears. When you enter a transaction ID for a synchronous transaction the Synchronous Details page appears.

Network Monitor Security

Upon accessing the monitor, you can see a list of all transactions in the system. However, to see specific information about a transaction and to view transaction details, you must have permission to the service operation.

See “Setting Permissions to Service Operations” (Integration Broker).

Understanding Network Monitor Processing Status Information

The Network Monitor uses the same asynchronous and synchronous processing statuses as those that appear in the Service Operations Monitor.

See “Understanding Asynchronous Service Operations Statuses” (Integration Broker Service Operations Monitor)“Understanding Synchronous Service Operation Statuses” (Integration Broker Service Operations Monitor).

Prerequisites for Using the Integration Network Monitor

Note the following prerequisites for using the Network Monitor:

- You must set up Network Monitor options using the Network Monitor Setup page. The options you set on the page apply to your monitoring activity in the Network Monitor and the Transactional Tracker.

  Information about setting up these options is provided elsewhere in this chapter.

  See Setting Up the Integration Network Monitor.

- To track transactions with remote PeopleSoft nodes the remote nodes must be active and defined in the integration network.

  See Registering Nodes in the Network.

- The Synchronous Transactions provides links the Synchronous Details page for viewing logging data. To capture logging data for a service operation you must set the log level on the routing definition for the service operation.

  See Registering Nodes in the Network.

Common Elements Used in the Integration Network Monitor

Field or Control	*Description*
	Click the button to show sender and receiver details in one view, instead of on separate sender and receiver tabs. Click the button again to collapse the view.
Filter	Click the button to filter the fetched data using the filters selected.
Last Sync	The read-only field displays the date and time that the transaction data was refreshed.
Monitor Setup	Click the link to access the Network Monitor Setup page to modify setup options, such as the number of rows to return for a query.
Network Node List	Displays the active integration network nodes for which you can track activity.
Re-Sync Monitor Data	Click the button to refresh the fetched data. The system automatically refreshes the data when you initially access any of the pages in the Transactional Tracker and when you initially select a node in the Network Node List.

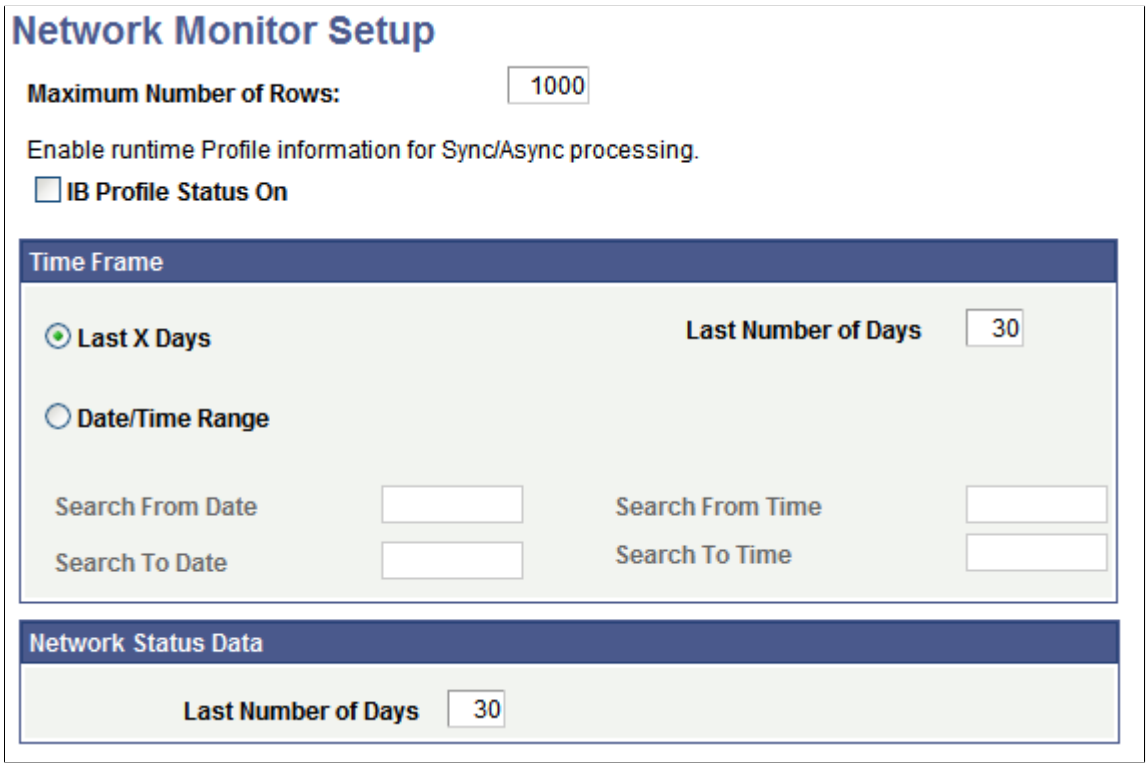
Field or Control	*Description*
Search Details	Click the link to retrieve a summary detail for a specific transaction. When you enter a transaction ID for an asynchronous transaction the Transactional Tracker Details page appears. When you enter a transaction ID for a synchronous transaction the Synchronous Details page appears.
Time Period	Use the options in the Time Period section to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range.

Setting Up the Integration Network Monitor

Use the Network Monitor Setup page (IB_INTNET_MONSETUP) to set up Integration Network Monitor and Transactional Tracker options, such as the number of rows to display in the monitor, date/time ranges for query results, and more.

To access the page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** and click the **Network Monitor Setup** link.

This example illustrates the fields and controls on the Network Monitor Setup page.



Specifying the Number of Rows to Return

Use the **Maximum Number of Rows** field on the Network Monitor Setup page to specify the number of rows to return as search results. The default value is *1000* rows.

There is no minimum or maximum value that you can set for this parameter. Consider integration volume when setting this value.

Enabling Integration Broker Performance Statistics

Check the **IB Profile Status On** field on the Network Monitor Setup page to enable the collection and display of runtime Integration Broker performance statistics.

Defining Date and Time Ranges for Query Results

Use the Time Frame section on the Network Monitor Setup page to define the date and time range in which query results should fall.

By default, the system returns search results for activity that has taken place in the past 30 days. However, you change the day range, or set parameters based on date and time ranges.

To define a date range in which query results are to fall:

1. Select the **Last X Days** radio button.
2. In the **Last Number of Days** field, enter a value.

   By default the value is *30,* and the system returns query results for the last 30 days of activity.

To define a date and time range in which query results are to fall:

1. Select the **Date/Time Range** radio button.
2. Select the date range.
   a. In the **Search from Date** field, enter the start date of the date range.

      Alternatively, click the calendar icon to select a date.

   b. In the **Search to Date** field, enter the end date of the date range.

      Alternatively, click the calendar icon to select a date.

3. Select the time range.
   a. In the **Search from Time** field, enter the start time of the time range.

   b. In the **Search to Time** field, enter the end time of the time range.

      Time field format is HH:MI:SS.999999, where HH represents hours, MI represents minutes, SS represents seconds, and 999999 represents microseconds.

Defining the Date Range of Network Status Query Results

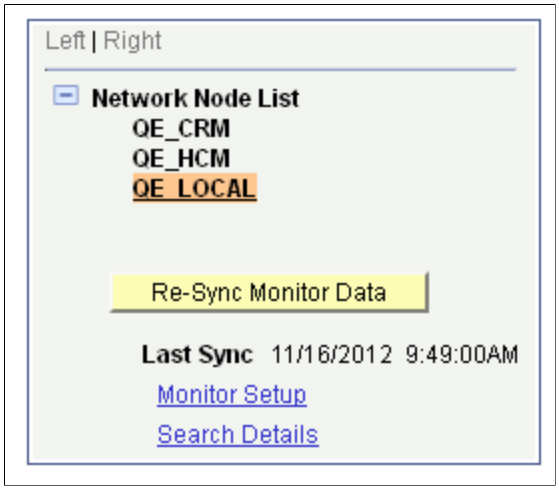
In the Network Status Data section, select the number of days for which to return results of network status information for the Network Status page.

The default value is *30* days.

Selecting Nodes to Monitor in the Integration Network Monitor

Each page of the Network Monitor features a **Network Node List.** The **Network Node List** contains a list of active integration network nodes for which you can monitor integration activity. The **Network Node List** appears on all of the Network Monitor pages.

This example illustrates the Network Monitor – Network Node List.



The example shows that there are three active network nodes for which you can monitor data, *QE_CRM,QE_HCM* and *QE_LOCAL.*. The *QE_LOCAL* node is highlighted and denotes that it is the currently selected node for monitoring network activity.

To change or select a node, click the node name in the tree.

When you select a node in the **Network Node List,** the system automatically resynchronizes and fetches the most current transaction data for the selected node.

Fetching and Synchronizing Integration Network Monitor Data

The PeopleSoft system fetches and synchronizes Network Monitor data when you initially access the Network Monitor component and when you change nodes in the **Network Node List.**

You can also manually synchronize data by clicking the **Re-Sync Monitor Data** button. The **Re-Sync Monitor Data** button appears on each page of the Network Monitor under the **Network Node List**.

The synchronization process synchronizes and updates the fetched data with changes in the database since the last synchronization.

A **Last Sync** field appears under the **Re-Sync Monitor Data** button and displays the date and time that the monitor data was last synchronized, whether it be by manual synchronization or system synchronization.

Filtering Integration Network Monitor Query Results

Controls on the Monitor Overview, Operation Instances, Publication Contracts, Subscription Contracts, and Synchronous Transaction pages provide data filters that enable you to narrow the fetched data results to that data which is most relevant to you.

The filters are listed on the pages on which they appear and are described in the Monitoring Asynchronous Integration Network Service Operations and Monitoring Synchronous Integration Network Service Operations in the following topics:

- “Common Elements Used to Filter and View Network Monitor Asynchronous Service Operation Data.”
- “Common Elements Used to Filter and View Network Monitor Synchronous Service Operation Data.”

After choosing the filters, you click the **Filter** button to obtain the filtered results.

Monitoring Asynchronous Integration Network Service Operations

This section discusses how to:

- Filter asynchronous network service operation data.
- Monitor general asynchronous network service operation data.
- Monitor asynchronous network service operation instances.
- Monitor asynchronous network service operation publication contracts.
- Monitor asynchronous network service operation subscription contracts.

Common Elements Used to Filter and View Network Monitor Asynchronous Service Operation Data

The following elements are used to filter and view the query results of Network Monitor service operation data.

Field or Control	*Description*
	Click the button to show sender and receiver details in one view, instead of on separate sender and receiver tab. Click the button again to collapse the view.
Alias	Service operation alias name if defined for the service operation.
Correlation ID	Identifier used to correlate messages sent separately into a single transaction.

Field or Control	*Description*
Creation Dttm	Creation date and time.
External Service Name	The name of the inbound service operation received from an integration partner. This name is equivalent to the routing alias.
Filter	Click the button to filter the fetched data using the filters selected.
Group By	Indicates how the system groups returned data. The valid values are: • *Queue.* (Default.) Displays results by queue name. • *Service Operation.* Displays results by service operation name.
Last Update Dttm	Last update date and time.
Orig Transaction ID	The original transaction ID generated and used for the service operation instance. As contracts are created another transaction ID is created for each publication or subscription contract. However, the original transaction ID is always available as a reference.
Publish, Publish Node, Node Name	Indicates the node that published the service operation.
Publish Dttm	Publish date and time.
Queue Level	Indicates the pub/sub queue for which to display information. The valid values are: • *OpInst.* (Default.) Operation instance. • *PubCon.* Publication contract. • *SubCon.* Subscription contract.
Queue Name	Name of the service operation queue.
Queue Sequence ID	Identifies the sequence of a particular service operation in a queue. This field is applicable to only service operations in ordered queues.

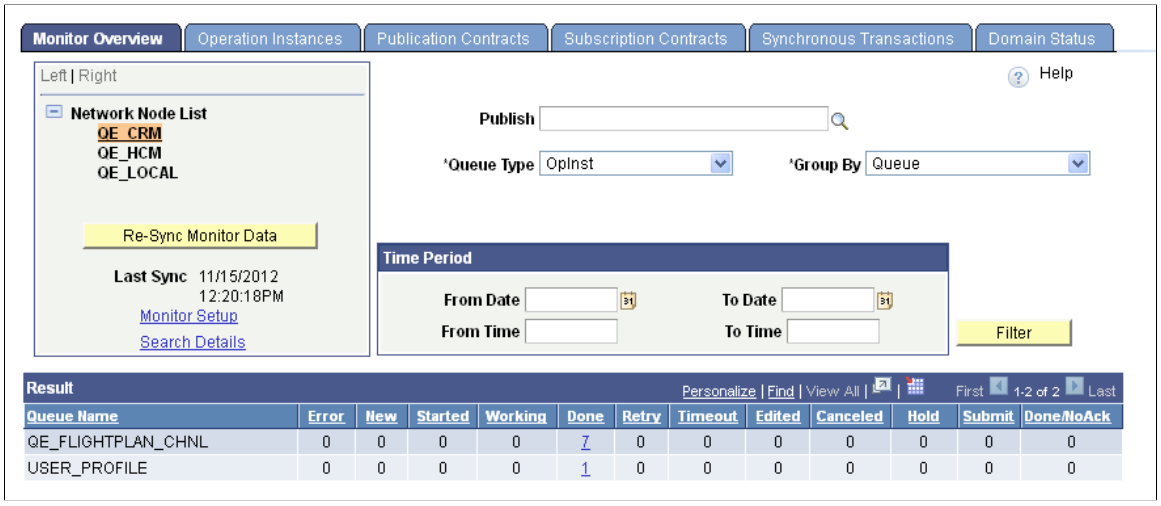
Field or Control	*Description*
Segment Number	When message segments are implemented, indicates the number of the segment message
Service Operation	Name of the service operation.
Service Operation Version, Version	Indicates the version of the service operation.
Status, Status String	Processing status of the service operation. The Network Monitor uses the same processing statuses as those that appear in the Service Operations Monitor. See “Understanding Asynchronous Service Operations Statuses” (Integration Broker Service Operations Monitor)“Understanding Synchronous Service Operation Statuses” (Integration Broker Service Operations Monitor).
Sub Node	The name of the subscribing node.
Sub Queue	If queue partitioning exists for a queue, a Sub Queue column appears in the Results grid on the Operation Instances page, Publication Contracts page and Subscription Contracts page . Click the link to open the Sub Queue Message Queue page to view all transactions in the sub queue.
Timestamp	Date and time of the publication.
Time Period	Use the options in the **Time Period** group box to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range. The **Time Period** group box features four fields for searching by date and time: **From Date, To Date, From Time** and **To Time.** If you complete just the date fields, the time fields automatically populate from *12:01 a.m.* to *11:59 p.m.* When left blank, no date or time is used as part of the search criteria.
Transaction ID	The unique identifier for a transaction.

Monitoring General Asynchronous Network Service Operation Information

Use the Monitor Overview page (IB_INTNET_OVRVIEW) for a high-level overview of the status of asynchronous service operation transactions in the integration network. You can group transactions

by queue or service operation for viewing. To access this page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor.**

This example illustrates the Network Monitor – Monitor Overview page.



When you access the page, the system automatically synchronizes the with the database for the selected node in the **Network Node List.** The processing status of asynchronous transactions appears in the Results grid at the bottom of the page. The number of operation instances in a particular status display as a linked value. Click the link to open the data in the Operation Instances page where you can view more detailed information.

If you change the node for which to view information, by clicking a different node name in the **Network Node List,** the system automatically resynchronizes the page with the most up-to-date transaction information for the node.

By default, the system queries the database by service operation queue for operation instance information.

You can query the database by modifying the query filters and then clicking the **Filter** button. The filters available on the page are:

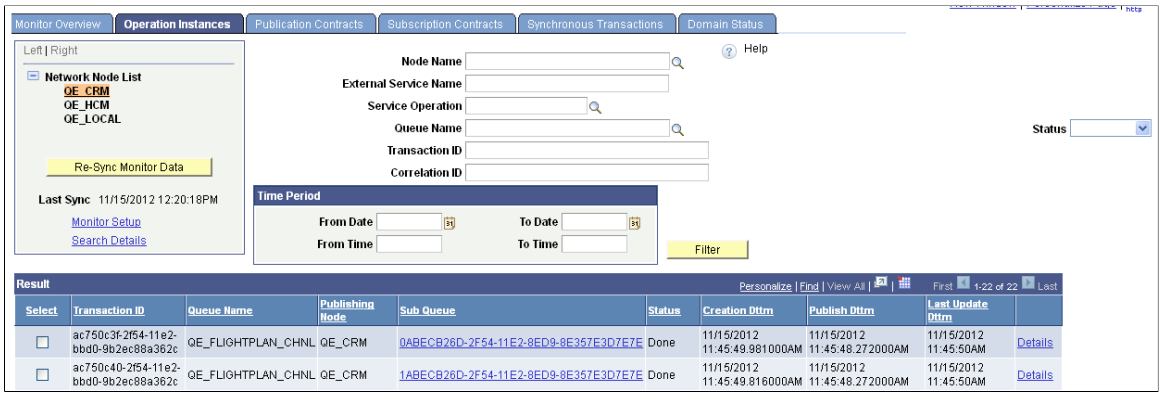
- **Publish.**
- **Query Level.**
- **Group By.**
- **Time Period.**

The data filters and data elements that appear in the Results grid are described in the Common Elements Used to Filter and View Asynchronous Network Monitor Data section of this topic.

Monitoring Asynchronous Network Service Operation Instances

Use the Operation Instances page (IB_INTNET_PUBHDR) to monitor the status and details related to individual asynchronous service operation instances. To access this page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor** and click the **Operation Instances** tab.

This example illustrates the Network Monitor – Operation Instances page.



The example shows a partial view of the query results appearing in the Results grid.

If you change the node for which to view information, by clicking a different node name in the **Network Node List,** the system automatically resynchronizes the page with the most up-to-date transaction information for the node.

By default, the system queries the database by service operation queue and the asynchronous processing status *Done.*

You can query the database by modifying the query filters and then clicking the **Filter** button. The filters available on the page are:

- **Node Name.**
- **External Service Name.**
- **Service Operation.**
- **Queue Name.**
- **Transaction ID.**
- **Correlation ID.**
- **Status.**
- **Time Period.**

Query results appear in the Results Grid. If the queue in which a transaction is processed is partitioned, a hyperlinked transaction ID appears in the Sub Queue column in the Results grid. Click the link to view all network service operations in the sub queue.

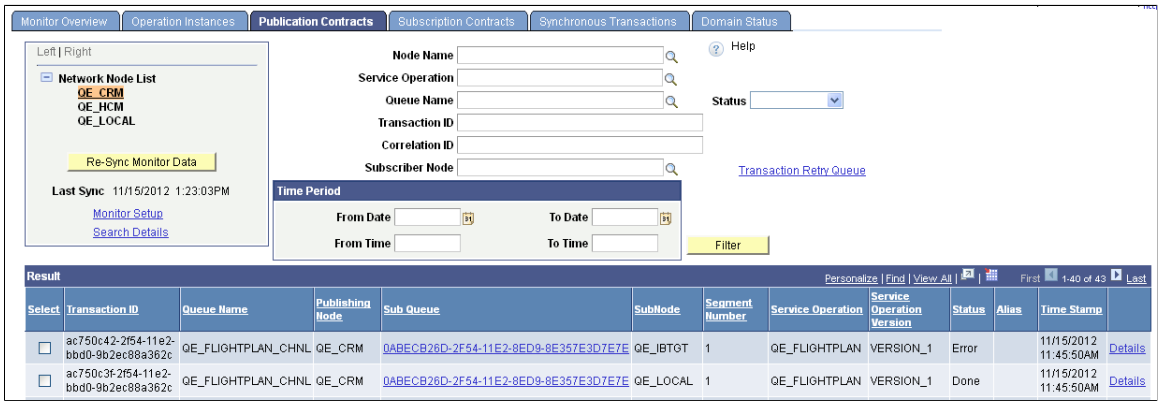
For each transaction in the Results grid a **Details** link appears. Click the **Details** link access the Asynchronous Details page and to view additional details for the transaction. Based on the processing status of the transaction, the Asynchronous Details page enables you to view any transaction error messages and resubmit or cancel the transaction.

The data filters and data elements that appear in the Results grid are described in the “Common Elements Used to Filter and View Asynchronous Network Monitor Data” section of this topic.

Monitoring Asynchronous Network Service Operation Publication Contracts

Use the Publications Contracts page (IB_INTNET_PUBCON) to monitor the status and details related to asynchronous service operation publication contracts. To access this page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor** and click the **Publication Contracts** tab.

This example illustrates the Network Monitor – Publication Contracts page. A partial view of the query results appear in the Results grid at the bottom of the example..



If you change the node for which to view information, by clicking a different node name in the **Network Node List,** the system automatically resynchronizes the page with the most up-to-date transaction information for the node.

By default, the system queries the database by service operation queue and the asynchronous processing status *Done.*

You can query the database by modifying the query filters and then clicking the **Filter** button. The filters available on the page are:

- **Node Name.**
- **Service Operation.**
- **Queue Name.**
- **Status.**
- **Transaction ID.**
- **Correlation ID.**
- **Subscriber Node.**
- **Time Period.**

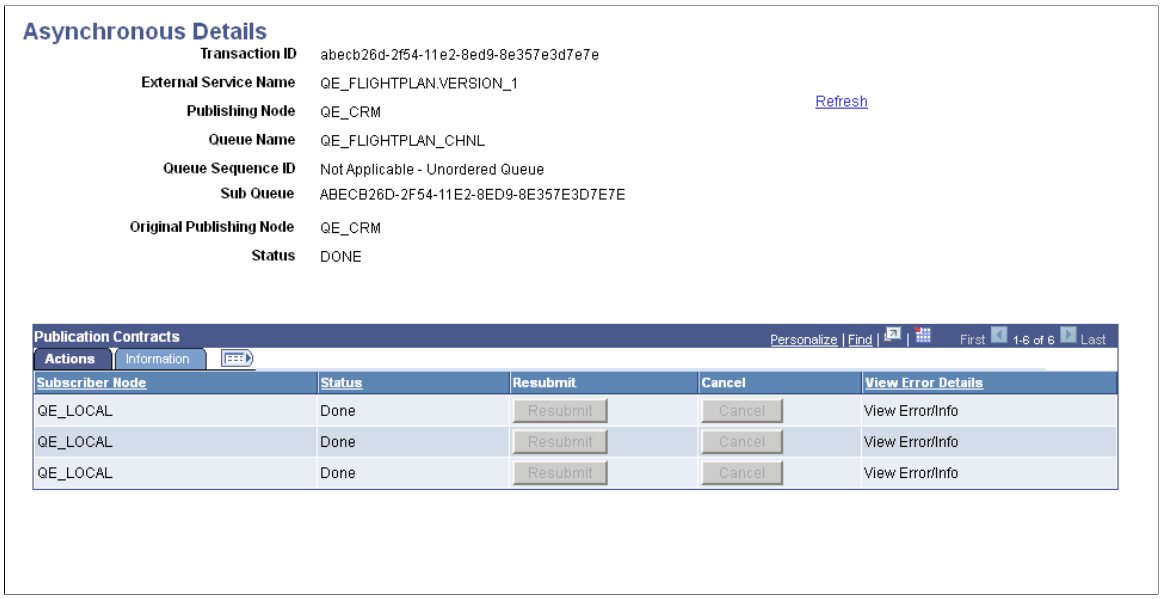
Query results appear in the Results grid.

The data filters and data elements that appear in the Results grid are described in the Common Elements Used to Filter and View Asynchronous Network Monitor Data section.

If the queue in which a transaction is processed is partitioned, a hyperlinked transaction ID appears in the Sub Queue column in the Results grid. Click the link to view all network service operations in the sub queue.

For each transaction in the Results grid a **Details** link appears. Click the **Details** link to access the Asynchronous Details page (IB_INTNET_DET) and to view additional details for the transaction. Based on the processing status of the transaction, the Asynchronous Details page enables you to view any transaction error messages and resubmit or cancel the transaction.

This example illustrates the details for an asynchronous network publication contract.



The top portion of the Asynchronous Details page shows the information that appears in the Results grid of the Publication Contracts page.

The Publication Contracts grid at the bottom of the page shows additional information about the transaction.

The Actions tab shows each node that is subscribing to the contract. In this example, the *QE_LOCAL* is subscribing to the contract. Depending on the processing status, the **Resubmit** and **Cancel** buttons are active and you can perform those related actions.

See Resubmitting and Cancelling Integration Network Transactions.

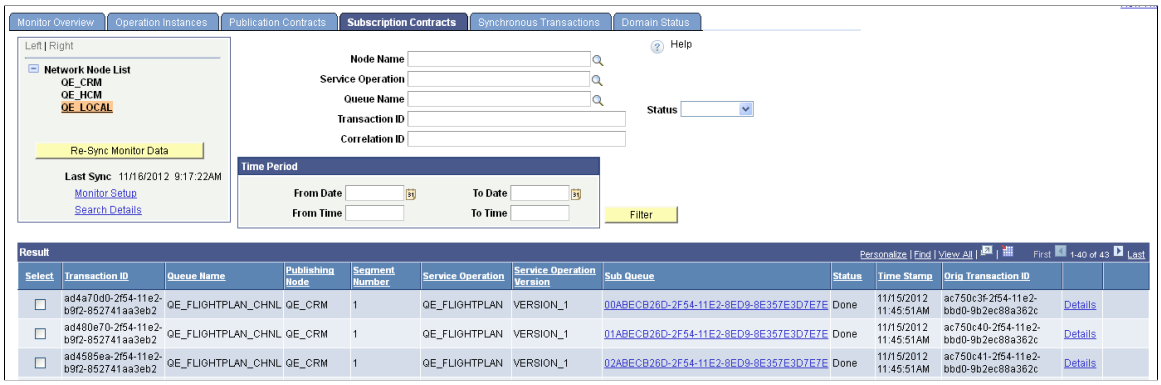
If there are any processing errors with a transaction, a **View Info Details** link appears. Click the link to view the error information. .

The Information tab reveals details about the publication transaction, including the transaction ID, the transaction time stamp, and so on.

Monitoring Asynchronous Network Service Operation Subscription Contracts

Use the Subscription Contracts page (IB_INTNET_SUBCON) to monitor the status and details related to asynchronous service operation subscription contracts. To access this page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor** and click the Subscription Contracts tab.

This example illustrates the Network Monitor – Subscription Contracts page.



If you change the node for which to view information, by clicking a different node name in the **Network Node List,** the system automatically resynchronizes the page with the most up-to-date transaction information for the node.

By default, the system queries the database by service operation queue and the asynchronous processing status *Done.*

You can query the database by modifying the query filters and then clicking the **Filter** button. The filters available on the page are:

- **Node Name.**
- **Service Operation.**
- **Queue Name.**
- **Status.**
- **Transaction ID.**
- **Correlation ID**
- **Time Period.**

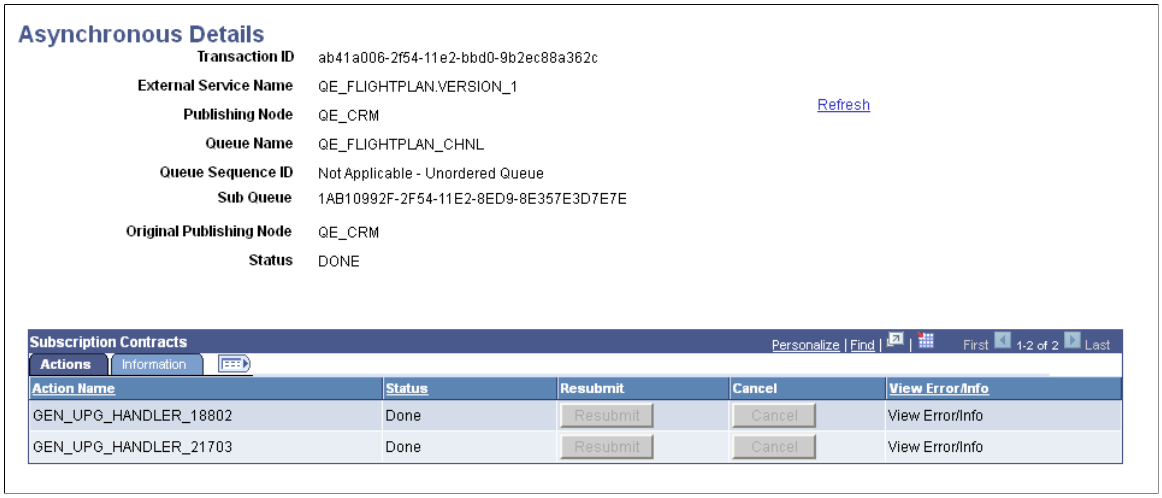
Query results appear in the Results grid.

The data filters and data elements that appear in the Results grid are described in the Common Elements Used to Filter and View Asynchronous Network Monitor Data section.

If the queue in which a transaction is processed is partitioned, a hyperlinked transaction ID appears in the Sub Queue column in the Results grid. Click the link to view all network service operations in the sub queue.

For each transaction in the Results grid a **Details** link appears. Click the **Details** link to access the Asynchronous Details page and to view additional details for the transaction. Based on the processing status of the transaction, the Asynchronous Details page enables you to view any transaction error messages and resubmit or cancel the transaction.

This example illustrates the Network Monitor – Asynchronous Details page for an asynchronous network subscription contract.



The top portion of the Asynchronous Details page shows the information that appears in the Results grid of the Subscription Contracts page.

The Subscription Contracts grid at the bottom of the page shows additional information about the transaction.

The Actions tab shows the service operation handler name for the subscription contract. Depending on the processing status, the **Resubmit** and **Cancel** buttons are active and you can perform those related actions.

See Resubmitting and Cancelling Integration Network Transactions.

If there are any processing errors with a transaction, a **View Info Details** link appears on the Actions tab. Click the link to view the error information.

The Information tab reveals details about the subscription transaction, including the transaction ID, the transaction time stamp, and so on.

Monitoring Synchronous Integration Network Service Operations

This section discusses how to monitor synchronous network service operation transactions.

Common Elements Used to Filter and View Network Monitor Synchronous Service Operation Data

The following elements are used to filter and view the query results of Network Monitor service operation data.

Field or Control	*Description*
Destination Publish Node	Identifies the name of the node where the service operation was sent.

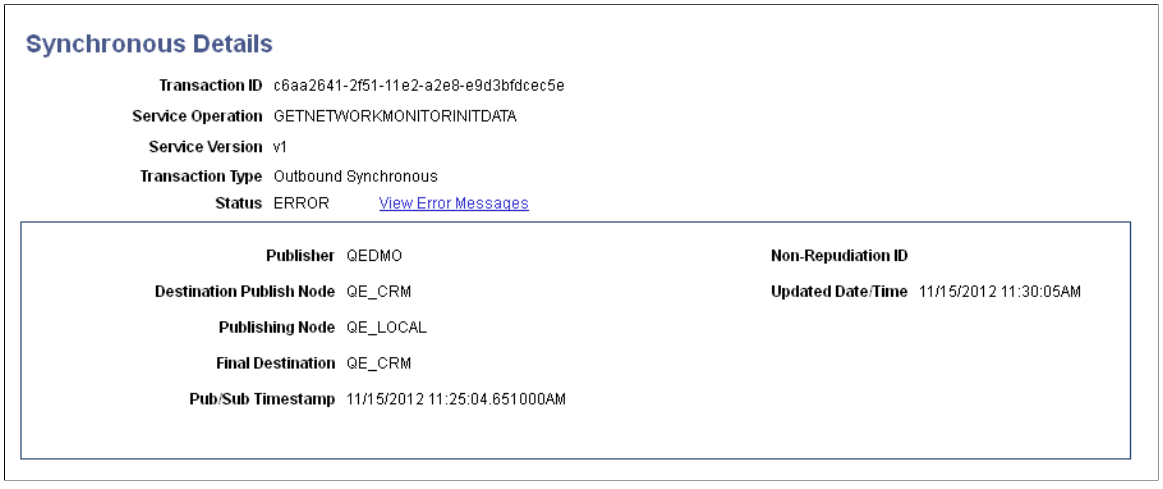
Field or Control	*Description*
External Service Name	The name of the inbound service operation received from an integration partner. This name is equivalent to the routing alias.
Filter	Click the button to filter the fetched data using the filters selected.
Final Destination	Identifies the name of the node of the final destination for the service operation.
Last Update Dttm	Last update date and time.
Non-Repudiation ID	Identifies a unique number used to associate a service operation instance with the nonrepudiation log.
Pub/Sub Timestamp	Identifies the date and time that the service operation instance was last processed.
Publish, Pub Node	Indicates the node that published the service operation.
Publish Dttm	Publish date and time.
Segment Number	When message segments are implemented, indicates the number of the segment message
Service Operation	Name of the service operation.
Service Operation Version, Version	Indicates the version of the service operation.
Status, Status String	Processing status of the service operation. The Network Monitor uses the same processing statuses as those that appear in the Service Operations Monitor. See “Understanding Asynchronous Service Operations Statuses” (Integration Broker Service Operations Monitor)“Understanding Synchronous Service Operation Statuses” (Integration Broker Service Operations Monitor).
Timestamp	Date and time of the publication.

Field or Control	*Description*
Time Period	Use the options in the **Time Period** group box to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range. The **Time Period** group box features four fields for searching by date and time: *From Date, To Date, From Time* and *To Time.* If you complete just the date fields, the time fields automatically populate from *12:01 a.m.* to *11:59 p.m.* When left blank, no date or time is used as part of the search criteria.
Trans Type	Indicates the type of synchronous transaction. The valid values are: • *OutSync*: Outbound Synchronous. • *InSync*: Inbound Synchronous.
Transaction ID	The unique identifier for a transaction.
Updated Date/Time	Identifies the date and time the service operation was last updated.

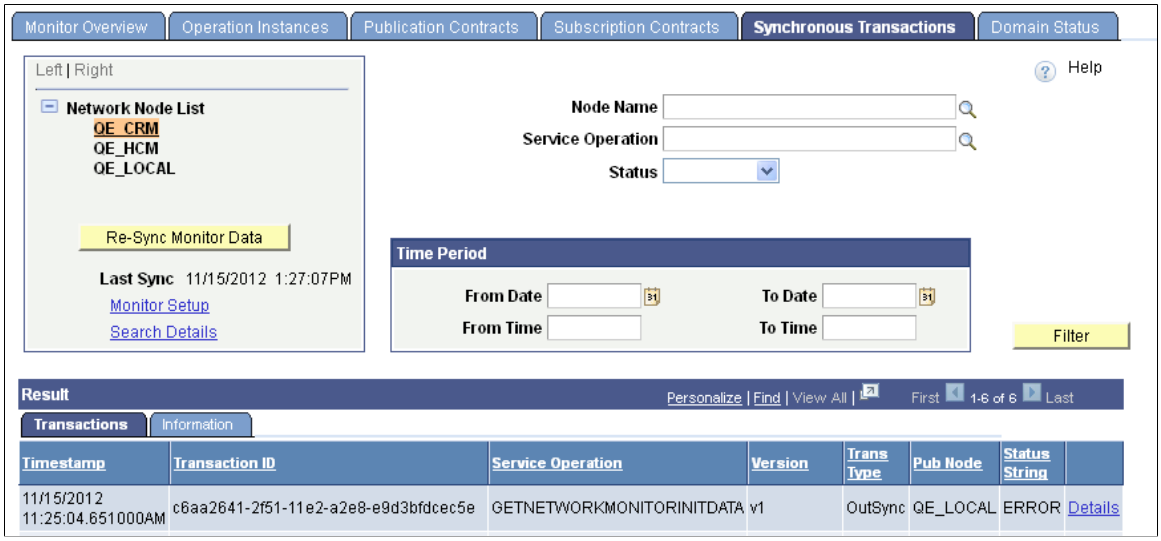
Monitoring Synchronous Network Service Operation Transactions

Use the Synchronous Transactions page (IB_INTNET_SYNC) to monitor the status and details related to synchronous service operations in the integration network. To access this page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor** and click the Synchronous Transactions tab.

This example illustrates the Network Monitor – Synchronous Details page.



This example illustrates the Network Monitor – Synchronous Transactions page. A partial view of the query results appears in the Results grid at the bottom of the example.



If you change the node for which to view information, by clicking a different node name in the **Network Node List,** the system automatically resynchronizes the page with the most up-to-date transaction information for the node.

You can query the database by modifying the query filters and then clicking the **Filter** button. The filters available on the page are:

- **Node Name.**
- **Service Operation.**
- **Status.**

Query results appear in the Results grid.

The data filters and data elements that appear in the Results grid are described in the Common Elements Used to Filter and View Synchronous Network Monitor Data section.

For each transaction in the Results grid a **Details** link appears. Click the **Details** link to access the Synchronous Details page (IB_INTNET_SYNC_DET) and to view additional details for the transaction. Based on the processing status of the transaction, the Asynchronous Details page enables you to view any transaction error messages and resubmit or cancel the transaction.

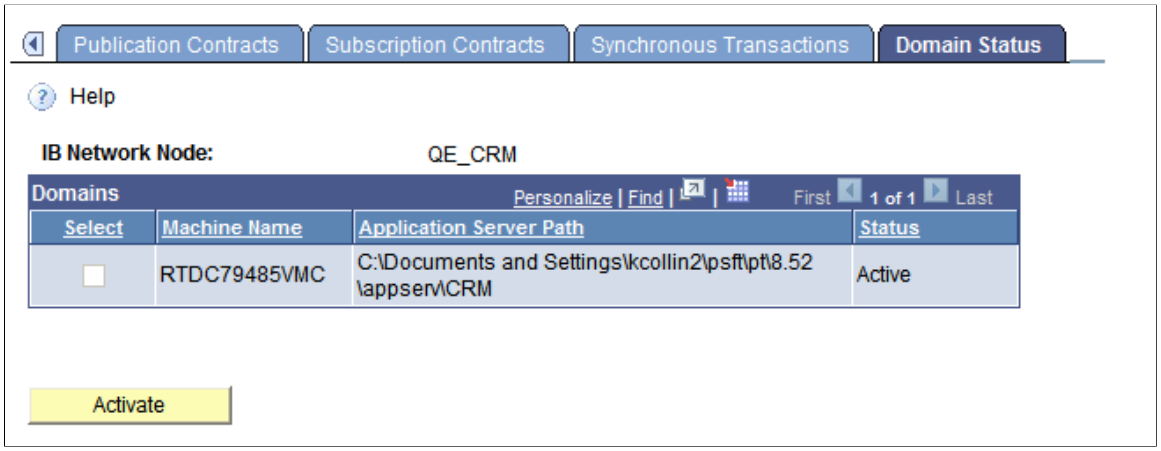
This example illustrates the Network Monitor – Synchronous Details page.

If there are any processing errors with a transaction, an **Error Messages** link appears, like the one shown in the example. Click the link to view error message information.

Monitoring Integration Network Domain Status

Use the Domain Status page (IB_INTNET_DOMSTAT) to monitor the status of the domain for the node selected in the **Network Node List.** To access this page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor** and click the Domain Status tab.

This example illustrates the Network Monitor – Domain Status page. Use this page to activate domains in the network.



Use the page to activate the domain of the selected node in the Network Node List, as necessary.

To activate a network node:

1. Access the Domain Status page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Monitoring** > **Integration Network Monitor** and click the Domain Status tab).

   The Network Monitor – Monitor Overview page appears.

2. In the **Network Node List,** select a node.
3. Click the **Domain Status** tab.
4. In the Domains grid, select the domain to activate.
5. Click the **Activate** button.

Resubmitting and Cancelling Integration Network Transactions

This section discusses how to:

- Resubmit or cancel individual network transactions for processing.
- Resubmit or cancel network transactions for processing in bulk.

Understanding Resubmitting and Canceling Network Transactions for Processing

You can resubmit and cancel network transactions for processing for only those to which you have permissions. If you attempt to resubmit or cancel a transaction for which you do not have permission, the system ignores the action.

Understanding Resubmitting and Canceling Future-Dated Network Transactions

You can cancel a future-dated transaction in the network monitor as you would any other transaction. However, when you cancel a future-dated transaction, the future date information is not retained. So if you subsequently resubmit the transaction, the system immediately submits it for processing.

Resubmitting and Canceling Individual Network Transactions

To resubmit or cancel individual transactions, select the check box next to the appropriate transaction and click the Resubmit or Cancel button. To deselect a transaction, clear the check box next to the transaction.

Resubmitting and Canceling Network Transactions in Bulk

You can resubmit and cancel network transaction for processing in bulk using the Network Monitor. Network transactions to resubmit in bulk must be in one of the following statuses:

- *Cancel.*
- *Edited.*
- *Error.*
- *Timeout.*

Network transactions to cancel in bulk must be in one of the following statuses:

- *Edited.*
- *Error.*
- *Hold.* (The transaction must also be future-dated.)
- *New.*

- *Retry.*
- *Timeout.*

In addition to the **Clear All, Resubmit** and **Cancel** buttons, you can also use the following links when resubmitting and canceling network transactions in bulk:

Field or Control	*Description*
Select All	Click the link to select all network transaction in the results grid to resubmit or cancel. After you click this link, click the **Resubmit** or **Cancel** button as appropriate.
Deselect All	Click the link to deselect all network transactions selected in the results grid.

Using the Integration Network Transactional Tracker

This section discusses how to:

- Select nodes to track in the Transactional Tracker.
- Fetch and synchronize Transactional Tracker data.
- Filter Transactional Tracker data.
- View network asynchronous transaction instances.
- View network asynchronous transaction detail information.
- View network synchronous transaction detail information.
- View network transaction information for specific transactions.

Understanding the Integration Network Transactional Tracker

The Network Transactional Tracker, or Transactional Tracker, enables you to track inbound and outbound integrations between the local node and all other nodes defined in the integration network.

The Transactional Tracker component (IB_INTNET_TRACKER) features three page for tracking activity:

Term	*Definition*
Tracker Overview	The Tracker Overview page (IB_INTNET_TRACKOVR) shows information about inbound and outbound asynchronous service operation instances, including the number of service operation instances, publication contracts, and subscription contracts that have been processed between the local node and a selected remote node. You can filter transaction information to display by queue or by service operation. For each transaction returned as a query result, the system provides a link to the Transactional Tracker Details page where you can review complete information for the transaction as well as cancel the transaction or resubmit the transaction for processing, depending on its processing status.
Tracker Main View	The Tracker Main View page (IB_INTNET_TRACKER) shows detailed information for inbound and outbound asynchronous transactions, including transaction ID, publishing timestamp, and more. You can filter transaction information by transaction ID, service operation name, queue name, and/or the direction of the transactions. This page features a link to the Integration Broker Statistics pages, so you can view transaction performance statistics to identify performance issues in the integration system. For each transaction returned as a query result, the system provides a link to the Transactional Tracker Details page where you can review complete information for the transaction as well as cancel the transaction or resubmit the transaction for processing, depending on its processing status.
Synchronous Tracker View	The Synchronous Tracker View page (IB_INTNET_TRACKSYN) enables you to view detailed information for inbound and outbound synchronous transactions between integration network nodes. You can filter transaction information by service operation name, the direction of the transaction, and/or by the status of processing status in the system. Like the Tracker Main View page, this page also features a link to the Integration Broker Statistics pages to view performance statistics and identify issues in the integration system. You can also launch the Integration Broker Service Operations Monitor – Synchronous Details page from this page to view error and logging information for specific transactions, depending on its processing status.

The pages in the Transactional Tracker feature links to the following pages that enable you to view details for a single transaction:

Term	*Definition*
Transactional Tracker Details	The Transactional Tracker Details page (IB_INTNET_TRACKDET) provides a snapshot view of all details for a single asynchronous transaction. The Tracker Main page and the Tracker Main View page provide links to the Transaction Tracker Details page. For the sending node and receiving node in the transaction, the system displays the transaction ID, the service operation name, queue name, processing status, and more. In addition, depending on the processing status you can cancel or resubmit the transaction for processing.
Synchronous Details	The Synchronous Details page (IB_INTNET_TRSYNCDT) provides a snapshot view of the transaction from the perspective of the sending node or the receiving node. The page features controls that enable you to view transaction errors and resubmit or cancel a transaction if there is a processing error.

Prerequisites for Using the Transactional Tracker

Note the following prerequisites for using the transactional tracker:

- You must set up Transaction Tracker monitor options using the Network Monitor Setup page. The options you set on the page apply to your monitoring activity in the Network Monitor and the Transactional Tracker.

  Information about options is provided elsewhere in the documentation.

- To track transactions with remote PeopleSoft nodes the remote nodes must be active and defined in the integration network.

- The Tracker Main View page and the Synchronous Tracker View page provide access to the Integration Broker performance statistics pages. To successfully use the performance statistics pages you must enable the statistics feature.

- The Synchronous Tracker View page provides links to the Synchronous Details page for viewing logging data. To capture logging data for a service operation you must set the log level on the routing definition for the service operation.

Related Links

Setting Up the Integration Network Monitor
Registering Nodes in the Network

Common Elements Used in the Transactional Tracker

Field or Control	*Description*
Network Node List	Displays the active integration network nodes for which you can track activity.

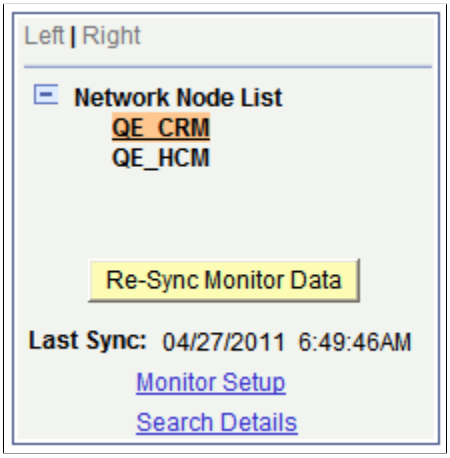
Field or Control	*Description*
Re-Sync Monitor Data	Click the button to refresh the fetched data. The system automatically refreshes the data when you initially access any of the pages in the Transactional Tracker and when you initially select a node in the Network Node List.
Last Sync	The read-only field displays the date and time that the transaction data was refreshed.
Monitor Setup	Click the link to access the Network Monitor Setup page to modify setup options, such as the number of rows to return for a query.
Search Details	Click the link to retrieve a summary detail for a specific transaction. When you enter a transaction ID for an asynchronous transaction the Transactional Tracker Details page appears. When you enter a transaction ID for a synchronous transaction the Synchronous Details page appears.
Time Period	Use the options in the **Time Period** group box to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range. The **Time Period** group box features four fields for searching by date and time: **From Date, To Date, From Time** and **To Time.** If you complete just the date fields, the time fields automatically populate from *12:01 a.m.* to *11:59 p.m.* When left blank, no date or time is used as part of the search criteria.
Filter	Click the button to filter the fetched data using the filters selected.
Sender Node	This read-only field displays the name of the sending node for the transaction query.
Receiver Node	This read-only field displays the name of the receiving node for the transaction query.
View Statistics	Click the link to view system performance statistics for asynchronous and synchronous transactions that flow through PeopleSoft Integration Broker. The statistics can help you to identify bottlenecks and other performance issues in your integration system.

Field or Control	*Description*
	Click the button to show sender and receiver details in one view, instead of on separate sender and receiver tab. Click the button again to collapse the view.
Sender	Click the tab to view transaction overview information for the sending node.
Receiver	Click the tab to view transaction overview information for the receiving node.

Selecting Nodes to Track in the Transactional Tracker

Each page of the Transactional Tracker features a **Network Node List.** The **Network Node List** contains a list of active integration network nodes for which you can track transaction activity between the local node. The **Network Node List** appears on all of the Transactional Tracker pages.

This example illustrates the Network Node List.



The example shows that there are two active network nodes, *QE_CRM* and *QE_HCM*. The *QE_CRM* node is highlighted and denotes that it is the currently selected node for tracking transactions between the local node.

To change or select a node, click the node name in the tree.

When you select a node in the **Network Node List,** the system automatically resynchronizes and fetches the most current transaction data for the selected node and the local node.

Fetching and Synchronizing Transactional Tracker Data

The PeopleSoft system fetches and synchronizes Transactional Tracker data when you initially access the Transactional Tracker component and when you change nodes in the **Network Node List.**

You can also manually synchronize data by clicking the **Re-Sync Monitor Data** button. The **Re-Sync Monitor Data** button appears on each page of the Transactional Tracker under the **Network Node List.**

The synchronization process synchronizes and updates the fetched data with changes in the database since the last synchronization.

A **Last Sync** field appears under the **Re-Sync Monitor Data** button and displays the date and time that the tracker data was last synchronized, whether it be by manual synchronization or system synchronization.

Filtering Transactional Tracker Data

Controls on each page the Transactional Tracker provide data filters that enable you to narrow the fetched data results to that data which is most relevant to you.

After choosing the filters, you click the **Filter** button to obtain the filtered results.

For example, each page of the Transactional Tracker features a **Time Period** group box where you can filter the fetched data by date or by date and time.

The specific data filters for each page are described in the Common Elements Used in the Transactional Tracker section and in the documentation for the pages on which they appear.

Viewing Network Asynchronous Transaction Instances

Use the Tracker Overview page (IB_INTNET_TRACKOVR) to view instance information for asynchronous transactions between the local node and remote nodes in the integration network.

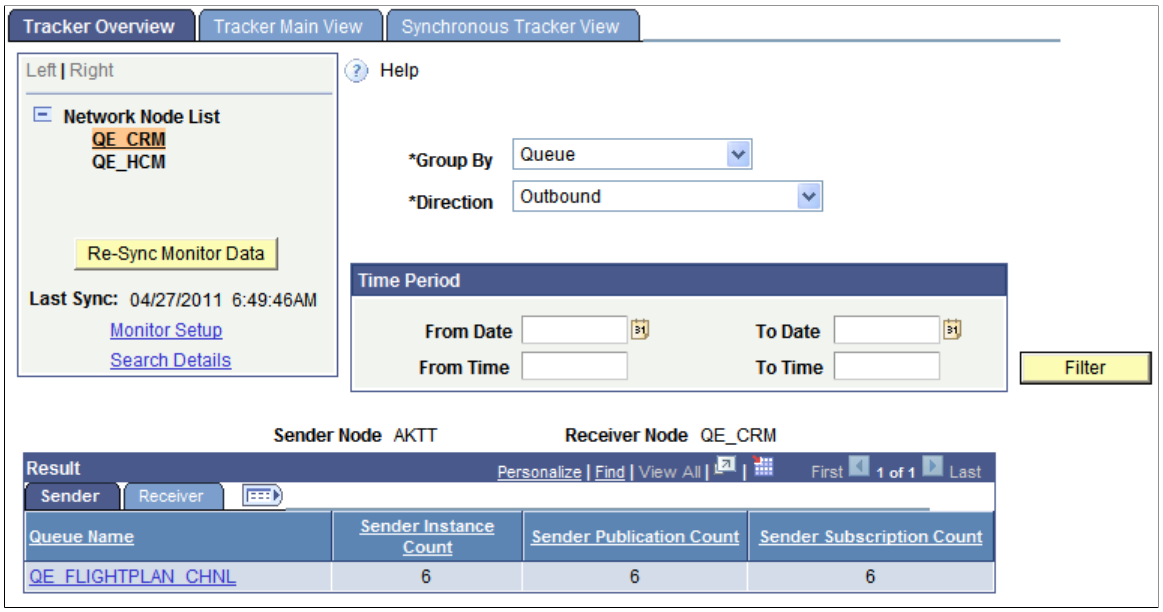
Before using the Tracker Overview it's a good idea to review the Prerequisites for Using the Transactional Tracker.

The page enables you to view transaction instance counts for sending and receiving nodes at each queue level in the system. You can view this information by service operation or queue, and the direction of the transaction (inbound or outbound).

When you access the Transactional Tracker component, the Tracker Overview page appears by default.

To access the Tracker Overview page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** and in the WorkCenter navigation pane, click **Transactional Tracker.**

This example illustrates the fields and controls on the Transactional Tracker – Tracker Overview page. You can find definitions for the fields and controls later on this page.



Use the **Network Node List** to select the remote node for which to track transactions. The system fetches available transaction data between the local node and the remote node. The system automatically synchronizes the data that appears on the page when you initially access the page and when you change the remote node.

Use one or more of the following data filters on this page:

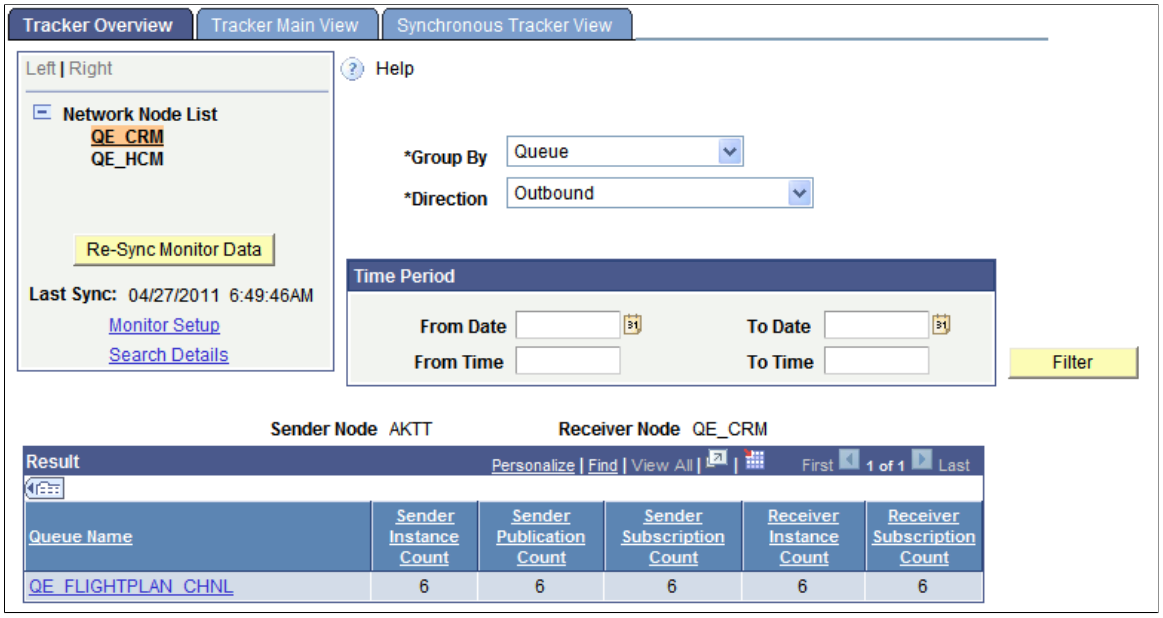
Field or Control	***Description***
Group By	Displays transactions based on the option selected. The valid options are: • *Queue.* (Default.) Use this option to filter transaction data by queue. Enter the queue name or click the Lookup button to search the database for a value. • *Service Operation.* Use this option to filter transaction data by service operation. Enter the service operation name or click the Lookup button to search the database for a value.
Direction	Indicates the direction of the transaction. The valid values are: • *Inbound.* Indicates an inbound transaction from the node selected in the Network Node List to the local node. • *Outbound.* (Default.) Indicates an outbound transaction from the local node to the node selected in the Network Node List.

Field or Control	*Description*
Time Period	Use the options in the **Time Period** group box to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range. The **Time Period** group box features four fields for searching by date and time: **From Date, To Date, From Time** and **To Time.** If you complete just the date fields, the time fields automatically populate from *12:01 a.m.* to *11:59 p.m.* When left blank, no date or time is used as part of the search criteria.

The Results grid displays the transaction instance data based on the default values for the filters. You can change the values for any of the fields and click the **Filter** button to display data based on the different filters.

When viewing the fetched data click the **Expand** icon to view the columns for all returned data.

This example illustrates the Tracker Overview page with the Results grid expanded to show all data returned by the system.



In the Results grid, click the Sender or Receiver tab to view counts of message instances, publication contracts, and subscription contracts processed by each respective node.

The following information appears in the Results grid on the Sender tab:

Field or Control	*Description*
Queue Name	Appears when the **Group By** filter is set to *Queue.* The name of the queue that contains the name of the instances shown. Click a queue name link to launch the Tracker Main View page to view details of the transactions associated with the queue.
Service Operation	Appears when the **Group By** filter is set to *Service Operation.* The name of the service operation that contains the instances shown. Click a service operation name link launch the Tracker Main View page to view details of the transactions associated with the service operation.
Sender Instance Count	Shows the number of message instances the sender has sent to the receiving node.

The following information appears in the Results grid on the Receiver tab:

Field or Control	*Description*
Sender Publication Count	Shows the number of publication contracts the sender has sent to the receiving node.
Sender Subscription Count	Shows the number of subscription contracts the sender has sent to the receiving node.
Receiver Instance Count	Shows the number of message instances received by the receiving node. When viewing the Sender tab you may have to click the Expand icon on the Results grid to view this field.
Receiver Subscription Count	Shows the number of subscription contracts received by the receiving node.

Viewing Network Asynchronous Transaction Detail Information

This section describes how to use the Transactional Tracker to view asynchronous transaction details.

Understanding Integration Network Asynchronous Processing Status Information

Asynchronous transactional detail information that the Transactional Tracker provides includes processing status information for service operation instances, publication contracts, and subscription contracts.

The processing statuses that the Transactional Tracker uses for asynchronous service operations are the same as those used in the Service Operations Monitor.

See “Understanding Asynchronous Service Operations Statuses” (Integration Broker Service Operations Monitor).

Viewing Integration Network Asynchronous Transactional Details

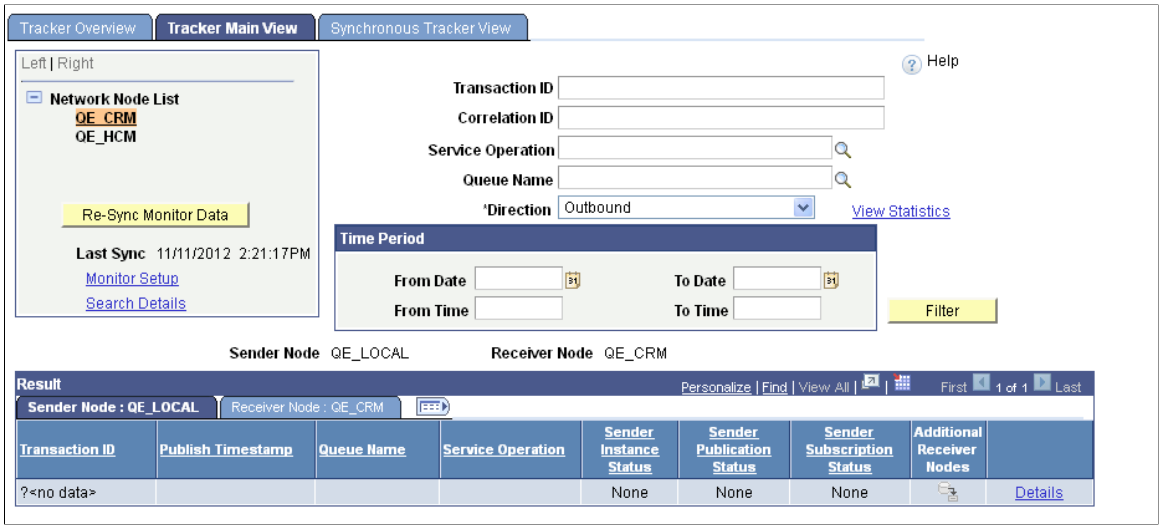
Use the Tracker Main View page (IB_INTNET_TRACKER) to view details for asynchronous transactions between the local node and remote nodes in the integration network.

Before using the Tracker Main View page it’s a good idea to review the Prerequisites for Using the Transactional Tracker.

The Tracker Main View page enables you to view the transaction IDs, queue names, processing status, and more for the transactions.

You can access the Tracker Main View page from the Tracker Overview page using the Results grid; click the queue name or service operation link in the grid. You can also access the page using the standard PeopleSoft Pure Internet Architecture navigation by selecting **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter.** In the WorkCenter navigation pane, click the **Transactional Tracker** link and then click the **Tracker Main View** tab.

This example illustrates the fields and controls on the Transactional Tracker – Tracker Main View page. You can find definitions for the fields and controls later on this page.



If you access the page from the Tracker Overview page, the page populates with data based on the query results from the Tracker Overview page. If you access the page without first conducting a query on the Tracker Overview page, click the **Re-Sync Monitor Data** button to populate the page with query data.

Use one of more of the following filters on this page:

Field or Control	***Description***
Transaction ID	Enter the unique identifier for the transaction in the PeopleSoft system.

Field or Control	*Description*
Correlation ID	Enter the unique identifier that is used to correlate messages that are sent separately into a single transaction.
Service Operation	Enter the name of a service operation to view transactions using the operation.
Queue Name	Enter the name of a service operation queue that is processing transactions to view.
Direction	Indicates the direction of the transaction. The valid values are: • *Inbound.* Indicates an inbound transaction from the node selected in the **Network Node List** to the local node. • *Outbound.* (Default.) Indicates an outbound transaction from the local node to the node selected in the **Network Node List.**
Time Period	Use the options in the **Time Period** group box to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range. The **Time Period** group box features four fields for searching by date and time: **From Date, To Date, From Time** and **To Time.** If you complete just the date fields, the time fields automatically populate from *12:01 a.m.* to *11:59 p.m.* When left blank, no date or time is used as part of the search criteria.

If you access the Tracker Main View page from the Tracker Overview page, the system defaults the queue name or service operation name from the Tracker Overview page as a filter. In addition, the direction of the transaction as viewed on the Tracker Overview page is also defaulted on the page.

The following information appears in the Results grid on the Sender Node tab:

Field or Control	*Description*
Transaction ID	The unique identifier for the transaction in the PeopleSoft system.
Publish Timestamp	Date and time the transaction was published. (Outbound transactions.)
Sender Instance Status	Processing status of the instance on the sending system.

Field or Control	*Description*
Sender Publishing Status	Processing status of the publication contract on the sending system.
Sender Subscription Status	Processing status of the subscription contract on the sending system.
Additional Receiver Nodes	Click the link to view the names of other nodes in the network receiving the transaction.
Details	Click the **Details** link for a summary view of the transaction using the Transaction Tracker Details page. In addition a snapshot view of sending and receiving node details, you can also cancel and resubmit transactions for processing from the page.

After a query the following information appears in the Results grid on the Receiver Node tab:

Field or Control	*Description*
Receiver Instance Status	Processing status of the service operation instance on the receiving system.
Receiver Subscription Status	Processing status of the subscription contract on the receiving system.

Viewing Network Synchronous Transactional Details

This section describes using the Synchronous Tracker View page to view integration network synchronous transactional details.

Understanding Integration Network Synchronous Processing Status Information

Synchronous transactional detail information that the Transactional Tracker provides includes processing status information for service operations.

The processing statuses that the Transactional Tracker uses for synchronous service operations are the same as those used in the Service Operations Monitor. The statuses for synchronous service operations are:

Field or Control	*Description*
Done	Indicates the synchronous request was successful.

Field or Control	*Description*
Error	Indicates that an error occurred during processing. Manual intervention is required.

See “Understanding Synchronous Service Operation Statuses” (Integration Broker Service Operations Monitor).

Prerequisites for Viewing Integration Network Synchronous Transactional Details

Before using the Synchronous Tracker View page see the Prerequisites for Using the Transactional Tracker.

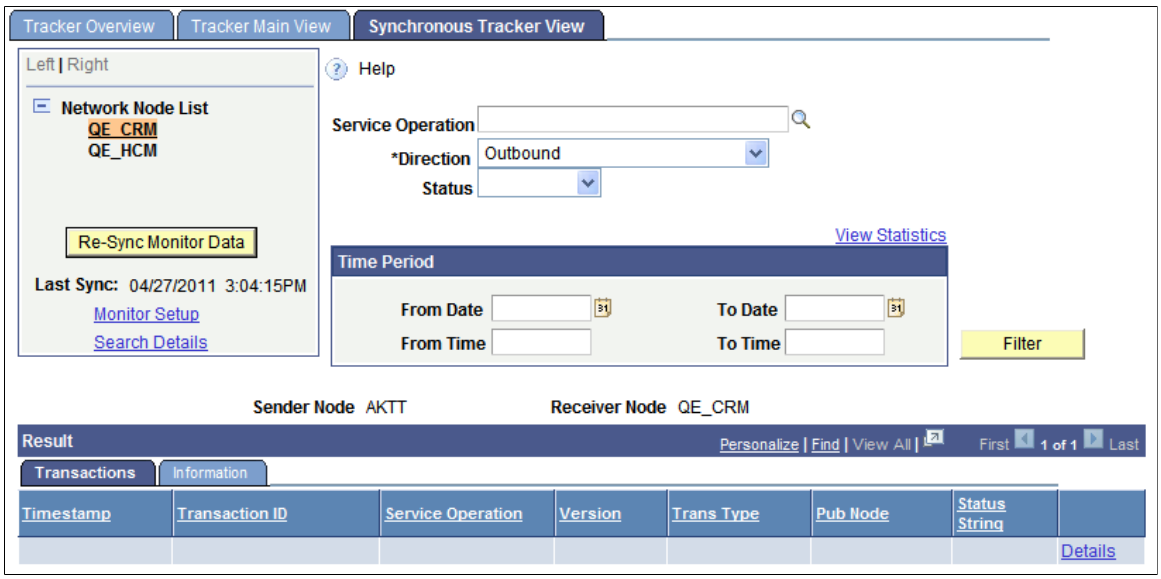
As part of the query results for synchronous transaction queries the system provides a **Details** link. The **Detail** link provides access to the Synchronous Details page in the Service Operations Monitor where you can view logging information. To access and view the logs you must set the logging level on the routing definition of all service operations for which you want to view logging information.

Viewing Integration Network Synchronous Transactional Details

Use the Synchronous Tracker View page (IB_INTNET_TRACKSYN) to view details for synchronous transactions between the local node and remote nodes in the integration network. The page enables you to view the transaction IDs, timestamps, service operation name, service operation version, service operation type, and more for integration network synchronous transactions.

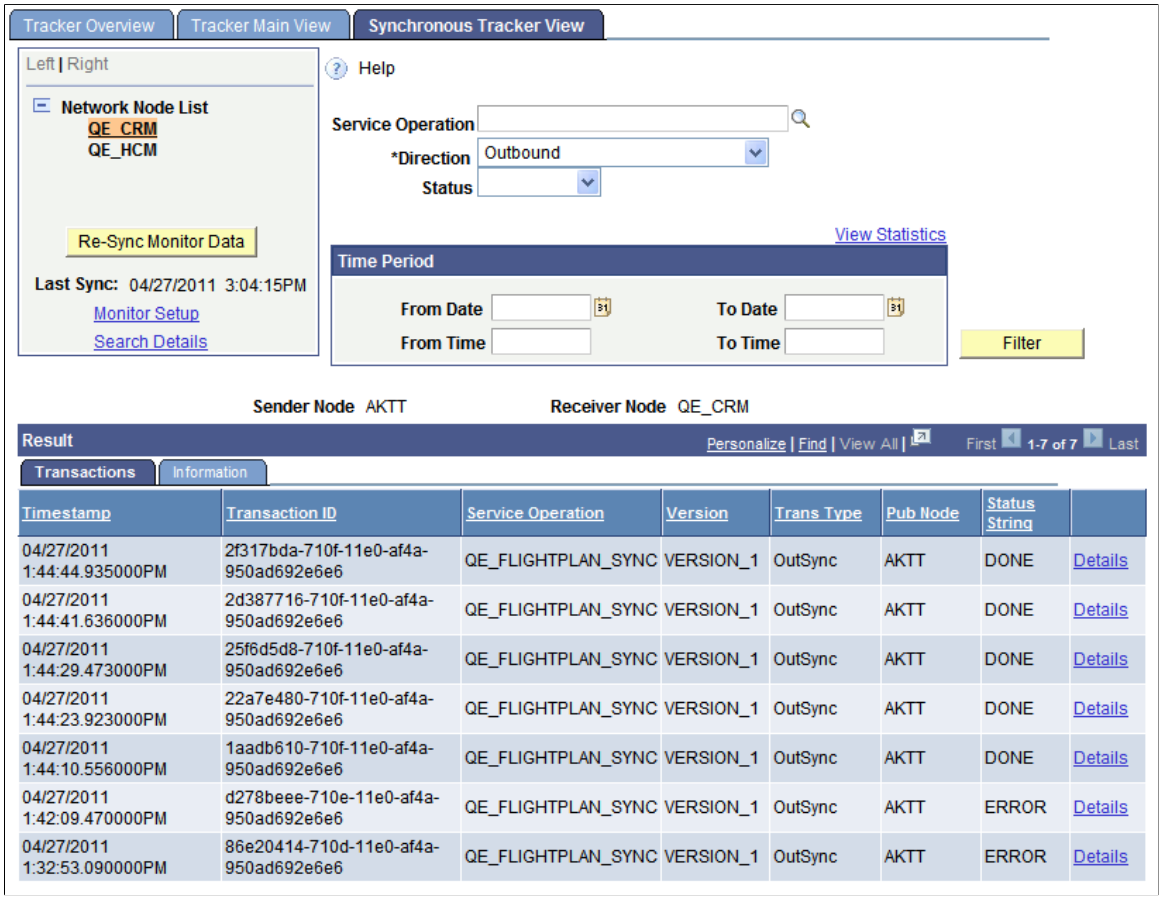
To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter.** Then in the WorkCenter navigation pane click the **Transactional Tracker** link, and click the **Synchronous Tracker View** page.

This example illustrates the fields and controls on the Transactional Tracker – Synchronous Tracker View page. You can find definitions for the fields and controls later on this page.



When you access the page, click the **Re-Sync Monitor Data** button fetch the data. The system displays the query results in the Results grid, based on the default filters.

This example illustrates query results for the *QE_CRM* node based on the default *Outbound* direction filter and clicking the **Re-Sync Monitor Data** button



As with all the pages in the Transactional Tracker, the Synchronous Tracker View page features several filter options so that you can view the most relevant data for your business requirements. After you select filter options, click the *Filter* button to apply the filter(s). View the results in the Results grid. The valid data filters for the Synchronous Tracker View page are:

Field or Control	***Description***
Service Operation	Use this option to filter transaction data by service operation. Enter the service operation name or click the **Lookup** button to search the database for a value.
Direction	Indicates the direction of the transaction. The valid values are: • *Inbound.* Indicates an inbound transaction from the node selected in the Network Node List to the local node. • *Outbound.* (Default.) Indicates an outbound transaction from the local node to the node selected in the Network Node List.

Field or Control	*Description*
Status	Filter the fetched results by the processing status of the service operation. The valid values are: • *Done.* • *Error.*
Time Period	Use the options in the **Time Period** group box to limit the query results to transactions that occurred during a specific date range or that occurred between a specific date and time range. The **Time Period** group box features four fields for searching by date and time: **From Date, To Date, From Time** and **To Time.** If you complete just the date fields, the time fields automatically populate from *12:01 a.m.* to *11:59 p.m.* When left blank, no date or time is used as part of the search criteria.

After a query the results appear on the Transactions and Information tabs in the Results grid.

The following information appears in the Results grid on the Transactions tab:

Field or Control	*Description*
Timestamp	The date and time the service operation was sent to the receiving node.
Transaction ID	The unique identifier for the transaction in the PeopleSoft system.
Service Operation	The name of the service operation.
Version	The service operation version.
Trans Type	The service operation type. The valid values are: • *OutSync:* Outbound synchronous. • *InSync:* Inbound synchronous.
Pub Node	The name of the node sending the transaction.

Field or Control	*Description*
String Status	The processing status of the service operation. The valid values are: • *Done.* • *Error.*
Details	Click the **Details** link to access the transaction in the Service Operations Monitor – Synchronous Details and view logging and error information. To capture logging information for a service operation you must set the logging level on the routing definition of the service operation. The Routings - Routing Definitions page features a **Log Details** drop-down list for setting the log level.

The following information appears in the Results grid on the Information tab:

Field or Control	*Description*
Publisher	Publisher of the service operation. This is usually the user ID of the person in the publishing system who triggered the publication.
Last Upd DtTm	Indicates the date and time the transaction was last updated.
NRID	Nonrepudiation ID. Identifies a unique number used to associate a service operation instance with the nonrepudiation log.
Dest Pub Node	Destination publish node. Identifies the name of the node where the service operation was sent.
Final Dest Node	Final destination node. Identifies the name of the node of the final destination for the service operation.
Details	Click the **Details** link to access the transaction in the Service Operations Monitor – Synchronous Details and view logging and error information. To capture logging information for a service operation you must set the logging level on the routing definition of the service operation. The Routings - Routing Definitions page features a **Log Details** drop-down list for setting the log level.

Related Links

"Defining General Routing Information" (Integration Broker)

Viewing Network Transaction Information for Specific Transactions

This section discusses how to:

- Search for transaction data.
- Access transaction data from query results grids.
- View data for an asynchronous transaction.
- View data for a synchronous transaction.

Understanding Viewing Integration Network Transaction Information for Specific Transactions

The Transactional Tracker features pages for viewing the details of specific transactions. Use the Transactional Tracker Details page to view information for an asynchronous transaction; use the Synchronous Details page to view information for a synchronous transaction.

Before using the Transactional Tracker Details page review the Prerequisites for Using the Transactional Tracker.

Searching for Transaction Data

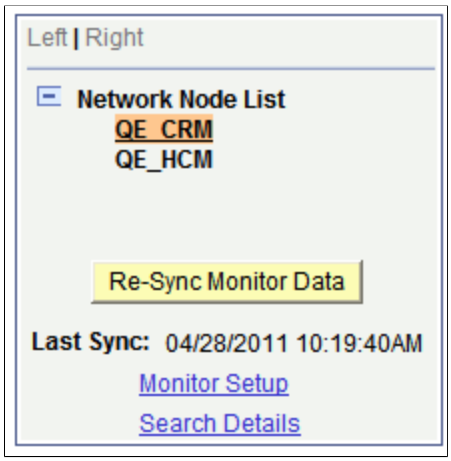
You can search for and access the Transactional Tracker Details page or the Synchronous Details page using the **Search Details** link that appears on the Tracker Overview page, the Tracker Main View page, and the Synchronous Main View page.

Access the pages as follows:

- Tracker Overview page: **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Transactional Tracker.**PeopleTools, Integration Broker, Integration Network WorkCenter, Transactional Tracker. The Tracker Overview page appears.
- Tracker Main View page: **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Transactional Tracker.** Click the Tracker Main View tab.
- Synchronous Main View page: **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Transactional Tracker.** Click the Synchronous Main View tab.

On each these pages the **Search Details** link appears under the **Network Node List.**

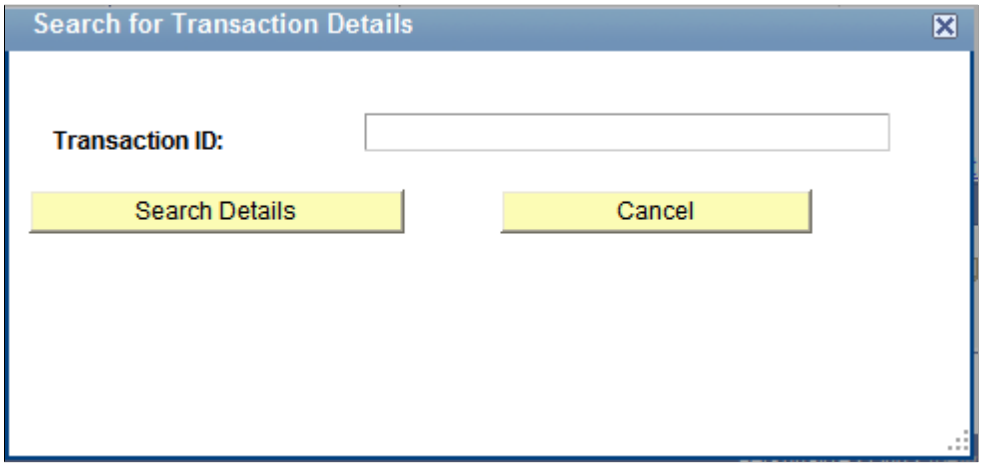
This example illustrates the **Search Details** link under the Network Node list



When you click the **Search Details** link the Search for Transaction Details page (IB_VERIFYNET_SEC) appears where you enter the transaction ID of the transaction you want to view.

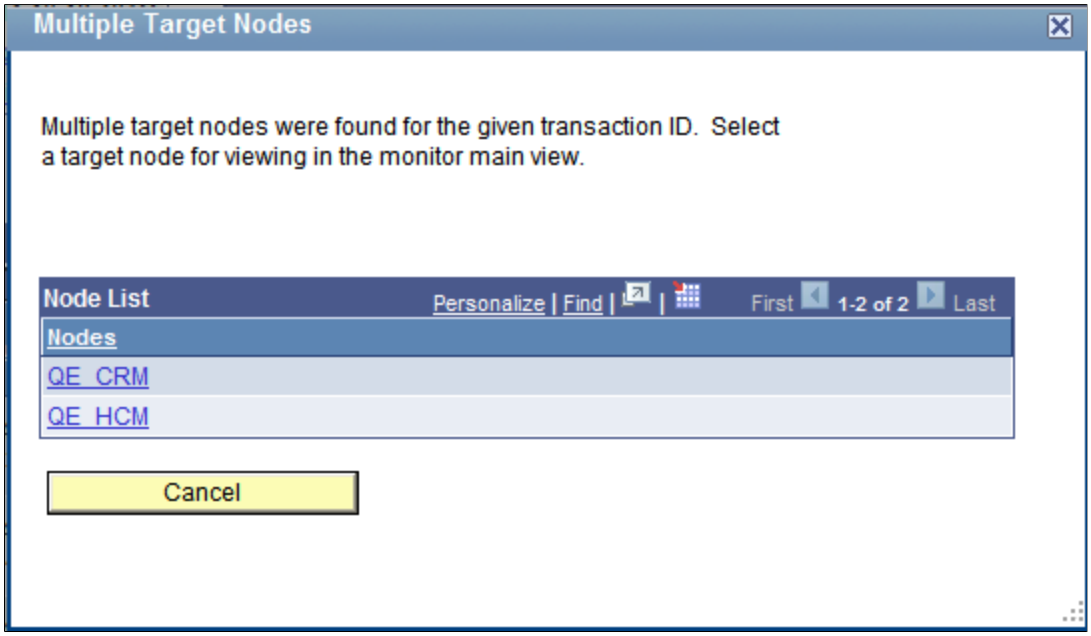
Important! The transaction ID that you enter should always be the instance transaction ID.

This example illustrates the Search for Transaction Details page.



When you click the **Search Details** button to search for the transaction the Multiple Target Nodes page (IB_NODELIST_SEC) appears if the transaction exists on more than one network node:

This example illustrates the Multiple Target Nodes page. In this example, a transaction exists on the nodes listed in the Node List grid, the *QE_CRM* and *QE_HCM* nodes.



If the transaction exists on more than one network node, click the name of the node for which you want to view the transaction.

If you selected an asynchronous transaction, the Transactional Tracker Detail page appears in a separate browser window for the transaction on the selected node. If you selected a synchronous transaction, the Synchronous Details page appears in a separate browser window for the transaction on the selected node.

Note: If the Transactional Tracker Detail page or the Synchronous Details page do not appear in a full-size browser window, click the browser **Restore** button in the upper right-corner of the browser to restore the browser window to full size.

Accessing Transaction Data from Query Results Grids

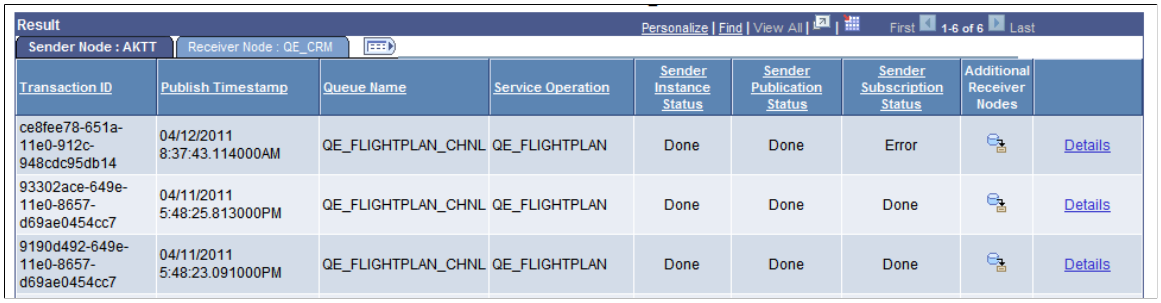
When you conduct a query on the Tracker Main View page or the Synchronous Main View page the results appear in the Results grid at the bottom of the page. At the end of each row of results is a **Details** link that provides access to the transaction details for the specific transaction.

Access the pages as follows:

- Tracker Main View page: **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Transactional Tracker.** Click the Tracker Main View tab.
- Synchronous Main View page: **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Transactional Tracker.** Click the Synchronous Main View tab.

After you select a remote node and synchronize the data the query results appear in the Results grid at the bottom of the page.

This example illustrates a partial view of the Results grid for an asynchronous query on the Tracker Main View page.



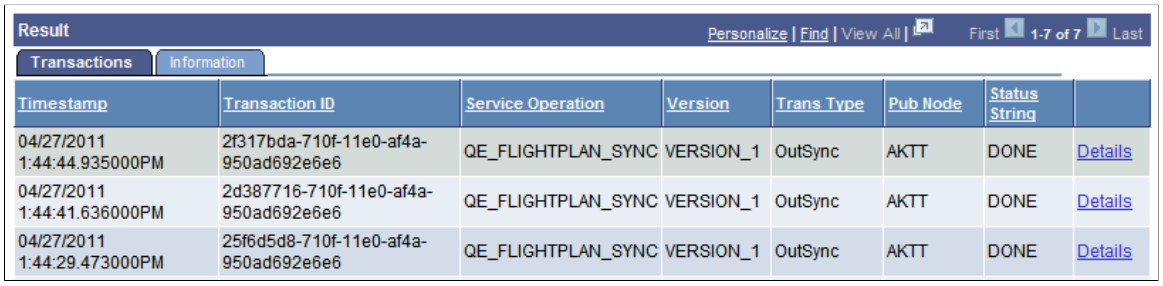
Result

Personalize | Find | View All | First 1-6 of 6 Last

Sender Node : AKTT | Receiver Node : QE_CRM

Transaction ID	Publish Timestamp	Queue Name	Service Operation	Sender Instance Status	Sender Publication Status	Sender Subscription Status	Additional Receiver Nodes	
ce8fee78-651a-11e0-912c-948cdc95db14	04/12/2011 8:37:43.114000AM	QE_FLIGHTPLAN_CHNL	QE_FLIGHTPLAN	Done	Done	Error		Details
93302ace-649e-11e0-8657-d69ae0454cc7	04/11/2011 5:48:25.813000PM	QE_FLIGHTPLAN_CHNL	QE_FLIGHTPLAN	Done	Done	Done		Details
9190d492-649e-11e0-8657-d69ae0454cc7	04/11/2011 5:48:23.091000PM	QE_FLIGHTPLAN_CHNL	QE_FLIGHTPLAN	Done	Done	Done		Details

This example illustrates a partial view of the Results grid for a synchronous query on the Tracker Main View page.



Result

Personalize | Find | View All | First 1-7 of 7 Last

Transactions | Information

Timestamp	Transaction ID	Service Operation	Version	Trans Type	Pub Node	Status String	
04/27/2011 1:44:44.935000PM	2f317bda-710f-11e0-af4a-950ad692e6e6	QE_FLIGHTPLAN_SYNC	VERSION_1	OutSync	AKTT	DONE	Details
04/27/2011 1:44:41.636000PM	2d387716-710f-11e0-af4a-950ad692e6e6	QE_FLIGHTPLAN_SYNC	VERSION_1	OutSync	AKTT	DONE	Details
04/27/2011 1:44:29.473000PM	25f6d5d8-710f-11e0-af4a-950ad692e6e6	QE_FLIGHTPLAN_SYNC	VERSION_1	OutSync	AKTT	DONE	Details

Click the **Details** link for a transaction to launch the Transactional Tracker Details page for an asynchronous transaction or to launch the Synchronous Details page for a synchronous transaction. Note that either page appears in a separate browser window.

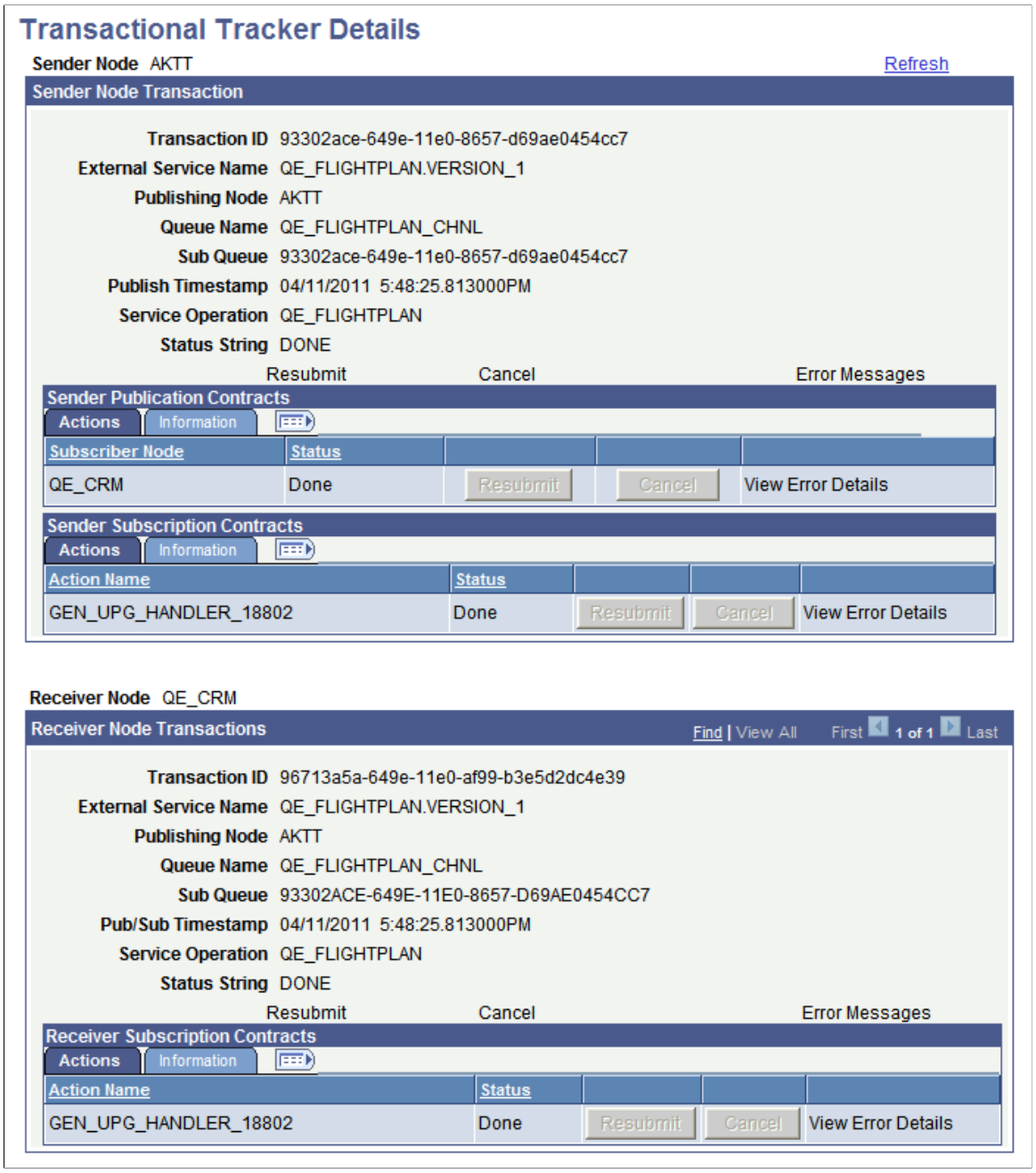
Note: If the Transactional Tracker Detail page or the Synchronous Details page does not appear in a full-size browser window, click the browser Restore button in the upper right-corner of the browser to restore the browser window to full size.

Viewing Data for an Asynchronous Transaction

The Transactional Tracker Details page features transaction information for the sending and receiving node.

To access the page click the **Details** link for a transaction in the Results grid on the Tracker Main View page.

This example illustrates the Transactional Tracker Details page.



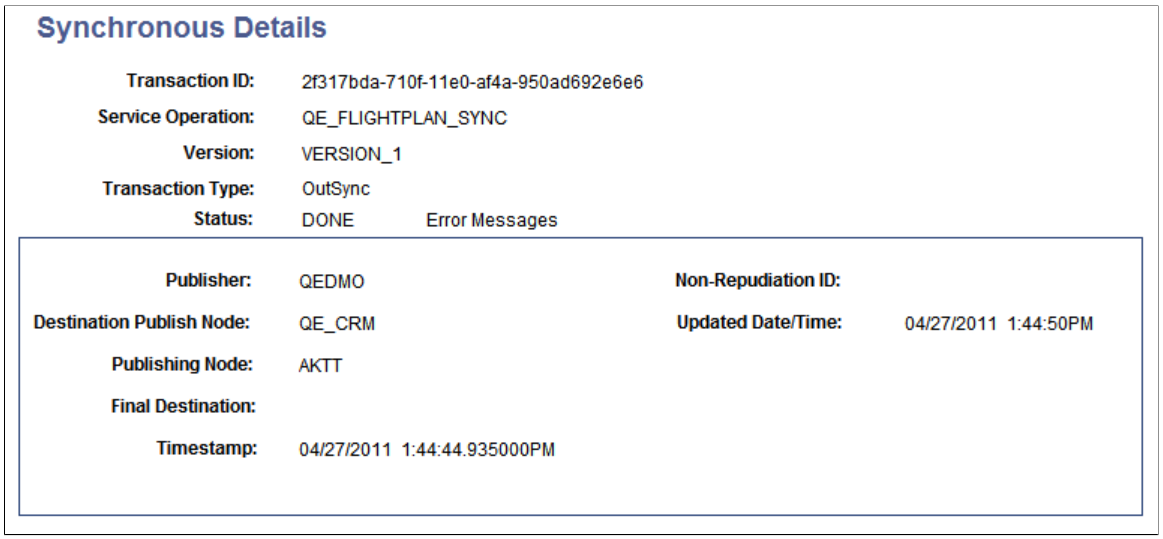
If errors exist in the transaction the **Error Messages** and **View Error Details** links are active and you can click them for additional details. The **Resubmit** and **Cancel** buttons on the page are enabled if the status of the transaction is such that these actions are warranted.

You can search for a specific transaction to view or you can launch the Transactional Tracker Details page from the results grid on the Tracker Main page or the Tracker Main View page.

Viewing Data for a Synchronous Transaction

The Synchronous Details page in the Transactional Tracker has the same format and content of the Synchronous Details page in the Integration Broker Service Operations Monitor, except that you cannot view transactional XML using the page in the Transactional Tracker.

This example illustrates the Synchronous Details page.



If errors exist in the transaction the **Error Messages** link is active and you can click the link for additional details. The **Resubmit** and **Cancel** buttons on the page are enabled if the status of the transaction is such that these actions are warranted.

The fields that appear on the page are described elsewhere in this section.

Performing Bulk Changes to Nodes

This topic discusses how to:

- Use the Integration Broker Network Bulk Change page.
- Search for definitions to change.
- Select and apply changes to node definitions.
- Select and apply changes to routing definitions.

Understanding Performing Changes to Nodes

PeopleSoft provides an Integration Broker Network Bulk Change page that enables you to make changes to one or more nodes at one time.

The utility enables you to search the database by one or more search options and then make changes to all or select nodes that match your search results.

You can use the utility to make changes to the following node properties:

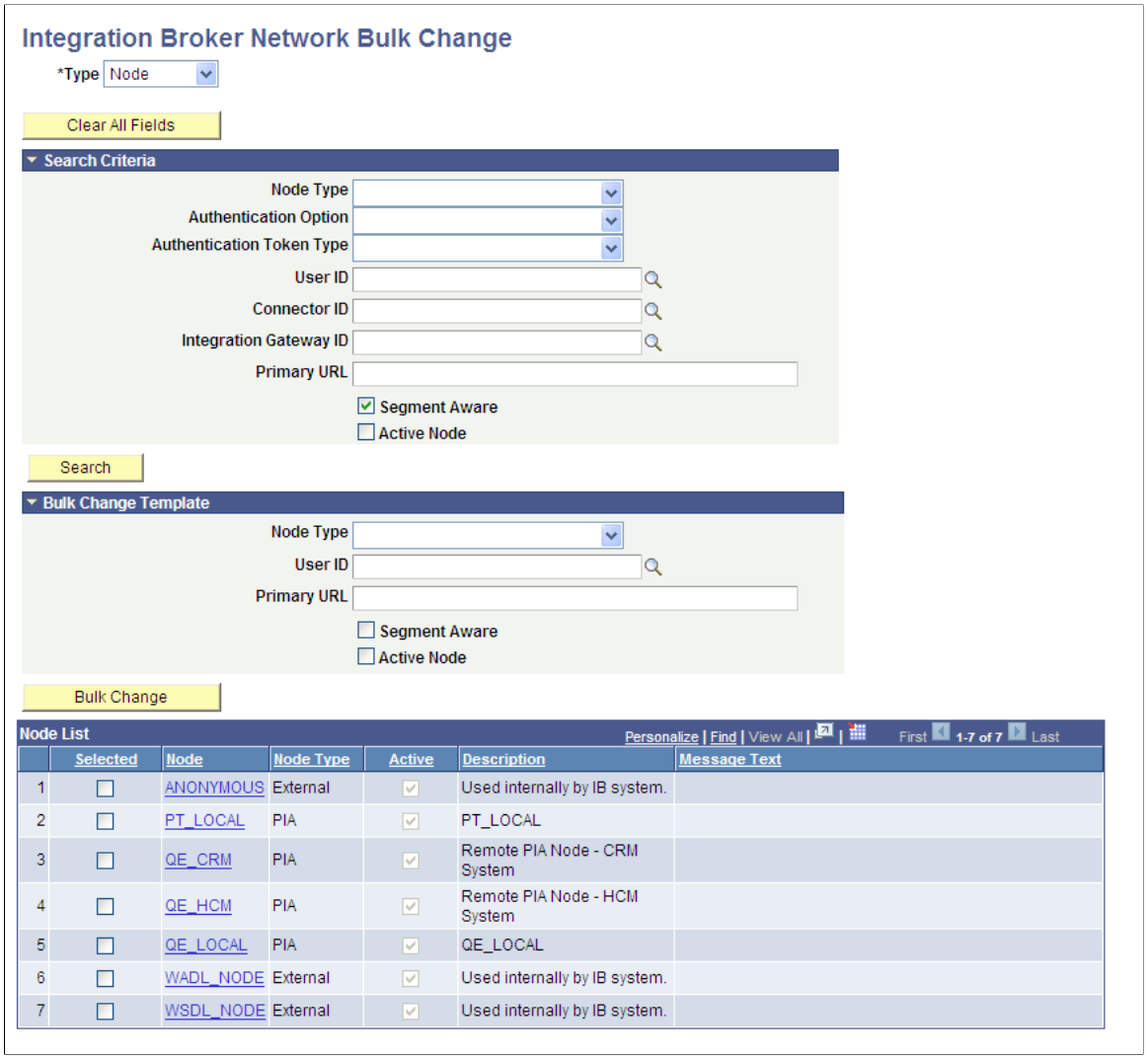
- Node type.
- User ID.
- Primary URL.

- Segment aware.
- Active node.

Using the Integration Broker Bulk Change page

To access the Integration Broker Network Bulk Change page (IB_BULKCHANGE) select **PeopleTools** > **Integration Broker** > **Integration Broker WorkCenter** > **Utilities** > **Network Node Bulk Change**

This example illustrates the fields and controls on the Integration Broker Network Bulk Change page when you are performing changes to node definitions. You can find definitions for the fields and controls later on this page.

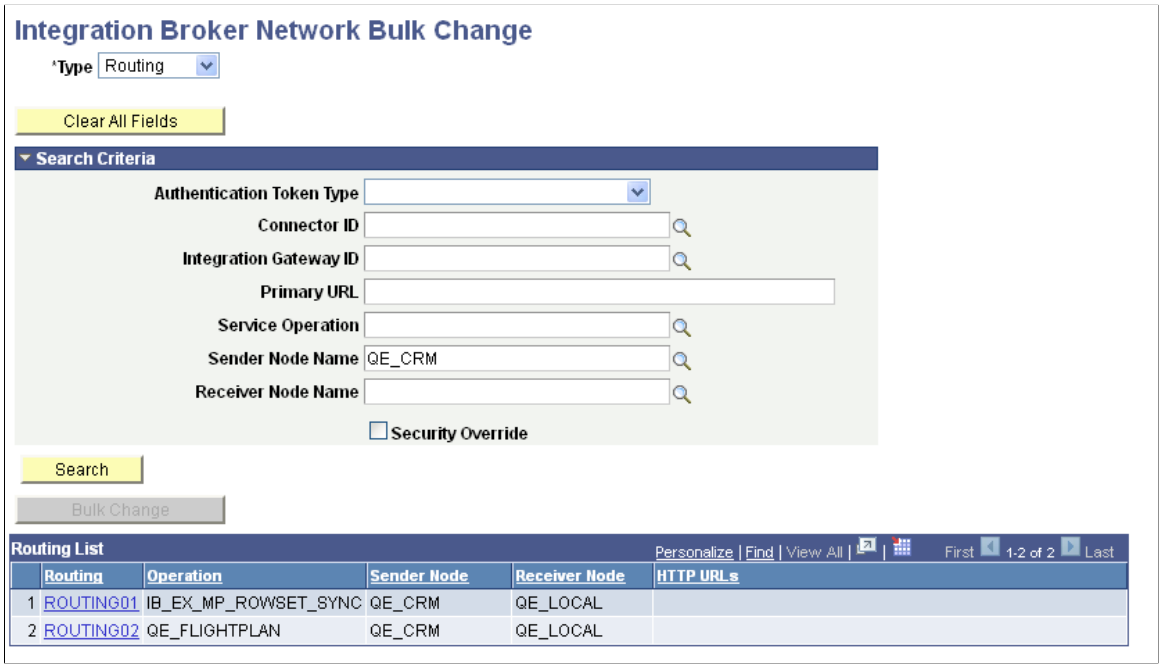


In the previous example, the Node List grid shows the results of searching the database for all nodes in the database that are segment aware.

You could now select any of the options in the Bulk Change Template section of the page, select the nodes in the Node List grid to which to apply the changes, and then click the Bulk Change button to apply the changes.

When you are performing changes to routing definitions, different fields appear on the Integration Broker Network Bulk Change page than when you are working with node definitions.

This example illustrates the fields and controls on the Integration Broker Network Bulk Change page when you are performing changes to routing definitions. You can find definitions for the fields and controls later on this page.



In the previous example, the Node List grid shows the results of searching the database for all routing definitions in the database where the sending node is *QE_CRM*.

You could now select any of the links that appear in the Routing List to make changes to the routing definition.

The following table describes the fields and controls on this page. Many of the fields are described in other sections in the product documentation. Links to existing documentation for fields and controls are provided at the end of table.

Field or Control	***Description***
Type	Indicates the definition type on which to perform changes. The values are: • *Node.* Select this option to make changes to node definitions in the database. • *Routing.* Select this option to make changes to routing definitions in the database.
Clear All Fields	Click the button to clear all fields in the Search Criteria and Bulk Change Template sections of the page.

Field or Control	*Description*
Node Type	Use this field to search for node definitions. In the Search Criteria section of the page, use this field to search for node definitions of one of the available types. In the Bulk Change Template section of the page, the value selected from the list will be applied as a change to selected nodes. Select a node type from the list. The values are: • *PIA.* • *External.* • *Pre-PeopleTools 8.4.*
Authentication Option	Use this field to search for node definitions. Search the database for node definitions that are defined with the authentication option you select. The options are: • *Certificate.* • *None.* • *Password.*
Authentication Token Type	Use this field to search for node definitions or routing definitions. Search the database for node definitions or routing definitions that are defined with the authentication token type you select. The options are: • *None.* • *SAML Token.* • *Username Token.* • *Username Token, no password.*
User ID	Use this field to search for node definitions. In the Search Criteria section of the page, enter a value in this field to search for nodes with the selected user ID. In the Bulk Change Template section of the page, the value selected from this field will be applied as a change to selected nodes.

Field or Control	*Description*
Connector ID	Use this field to search for node definitions or routing definitions. Enter a target connector ID. The system searches the database for node definitions or routing definitions that are defined with the connector ID.
Integration Gateway ID	Use this field to search for node definitions or routing definitions. Enter an integration gateway ID. The system will search the database for node definitions or routing definitions that are associated with the local default node.
Primary URL	Use this field to search for node definitions or routing definitions. In the Search Criteria section of the page, enter a value in this field to search for node definitions that have the primary URL value defined for the target connector associated with the node. If you are searching for routing definitions, enter the primary URL value defined for a target connector associated with a sending or receiving node defined on a routing definition. When performing changes to nodes, the **Primary URL** field that you enter in the Bulk Change Template section of the page is applied as a change to selected nodes.
Segment Aware	Use this field to search for node definitions. In the Search Criteria section of the page, select the box to search for node definitions that are segment aware or clear the box to search for node definitions that are not segment aware. In the Bulk Change Template section of the page, select the box or clear the box to apply or remove the segment aware status to selected node definitions.
Active Node	Use this field to search for node definitions. In the Search Criteria section of the page, select the box to search for node definitions that are active; clear the box to search for nodes that are not active. In the Bulk Change Template section of the page, select the box or clear the box to apply or remove the active status to selected node definitions.
Service Operation	Use this field to search for routing definitions. Enter the name of the service operation associated with the routing definition you want to modify.

Field or Control	*Description*
Sender Node Name	Use this field to search for routing definitions. Enter the name of the sending node associated with the routing definition you want to modify.
Receiver Node Name	Use this field to search for routing definitions. Enter the name of the receiving node associated with the routing definition you want to modify.
Security Override	Use this control to search for routing definitions. Check the box to search for routing definitions that have WS security overrides at the routing level.
Search	Click the button to search for node definitions or routing definitions based on the criteria entered in the Search Criteria section of the page.
Node List	Grid where the search results for node definitions appear. The fields and controls that appear in the grid are described later on in this section.
Routing List	Grid where the search results for routing definitions appear. The fields and controls that appear in the grid are described later on in this section.
Bulk Change	Click the button to apply the options in the Bulk Change Template section of the page to the selected nodes in the Node List.

See Configuring Nodes, Implementing Web Services Security, “Configuring Service Operation Definitions” (Integration Broker), “Configuring Routing Definitions” (Integration Broker), Understanding Target Connectors

The Node List grid displays results when you search for node definitions. The Node List grid features the following fields and controls:

Field or Control	*Description*
Selected	Select the box next to each node to which to apply changes.
Node	Indicates the node name. Click the link to access the definition for the node in the Nodes – Node Definition page. Click the **Return** button on the Nodes – Node Definition page to return to the Integration Broker Network Bulk Change page.

Field or Control	*Description*
Node Type	Indicates the type of node. The possible values are described in the previous table.
Active	This read-only box is selected when the node is active.
Description	Description for the node as defined in the node definition.
Message Text	After a bulk change operation, the results of the process appear in this column. This field is read-only.

The Routing List grid displays results when you search for routing definitions. The Routing List grid features the following fields and controls:

Field or Control	*Description*
Routing	Routing name. Click the link to access the definition for the routing in the Routings – Routing Definition page. Click the **Return** button on the Routings – Routing Definition page to return to the Integration Broker Network Bulk Change page.
Operation	Name of the service operation where the routing is used.
Sender Node	Name of the node that is defined to send the service operation.
Receiver Node	Name of the node that is defined to receive the service operation.
HTTP URLs	When the local node is the sending node, this field displays the endpoint of the integration when the HTTP target connector is defined as the connector on the Routings – Connector Properties page.

See “Configuring Routing Definitions” (Integration Broker)

Searching for Definitions to Change

To search for definitions to change:

1. Access the Integration Broker Network Bulk Change page (**PeopleTools** > **Integration Broker** > **Integration Broker WorkCenter** > **Utilities** > **Network Node Bulk Change**).
2. Select the type of definition to modify.

   From the **Type** drop-down list, select one of the following definition types:

- *Node.*
- *Routing.*
- To narrow search results, select one or more options in the Search Criteria section of the page as described previously in this topic, or
- Leave all fields blank to return a list of all definitions in the database for the selected definition type.

3. Click the **Search** button.
4. The results appear in a grid at the bottom of the page.

Selecting and Applying Changes to Node Definitions

To select and apply changes to node definitions:

1. In the Bulk Change Template section of the Integration Broker Network Bulk Change page, select the changes to apply.

   The fields and controls in the Bulk Change Template section are described previously in this topic.

2. In the Node List grid select the box next to each node to apply changes.
3. Click the **Bulk Change** button.

The Message Text column of the Node List grid displays the results of the action for each node selected.

You can also open the definition for a node by clicking the node name in the Node List grid. When you do so the Nodes – Node Definition page appears and you can view and modify the definition as appropriate. Click the **Return** button on the Nodes – Node Definition page to return to the Integration Broker Network Bulk Change page.

Selecting and Applying Changes to Routing Definitions

To select and apply changes to routing definitions:

1. In the Routing List grid, click the name of the routing to change.

   The routing definition appears in the Routing – Routing Definitions page.

2. Make the desired changes to the definition.
3. Save the changes to the definition.
4. Click the **Return** button to return to the Integration Broker Network Bulk Change page.

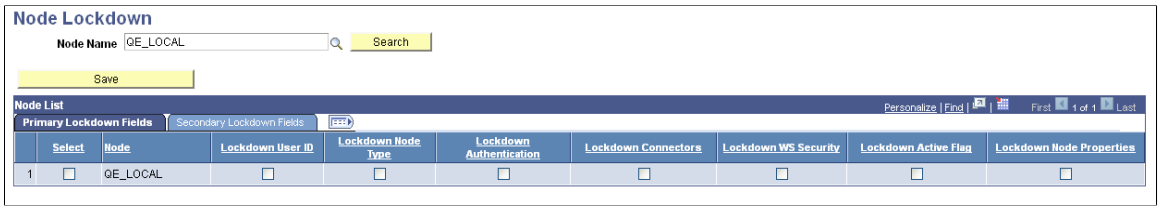
Locking Down Node Properties for Project Copy

The Integration Network features a Node Lockdown page that enables you to lock select node properties on nodes defined in the local database so that they are not overwritten during project copy.

The properties that the system locks are those defined for the node in the Node component.

To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Utilities** > **Lock Down Node Properties.**

This example illustrates the fields and controls on the Node Lockdown page. You can find definitions for the fields and controls later on this page.



The previous example shows the Node Lockdown page after searching the database for the *QE_LOCAL* node. You can search for and work with one node at a time by searching for a specific node, as this example illustrates. You can also return and work with a list of all nodes defined in the local database by leaving the **Node Name** field blank and clicking the **Search** button.

The Node List grid displays the search results and lists the node properties that you can lock.

Note that the Node List grid contains two tabs. The **Primary Lockdown Fields** grid contains node fields and properties that you can lock down such as the node user ID, node authentication, WS Security settings, and so on. The **Secondary Lockdown Fields** grid contains fields and properties you can lock down, including the default local node option, the segment aware option, and others.

This Node Lockdown page features the following fields and controls:

Field or Control	***Description***
Node Name	Enter or select a node name with which to work. Leave the field blank to display all nodes in the database.
Search	Click the button to search the database.
Save	After you select the node properties to lock, click the button to save the changes.
Primary Lockdown Fields	Click the tab to view and select primary node properties to lock
Secondary Lockdown Fields	Click the tab to view and select secondary node properties to lock

Field or Control	*Description*
Select	Select the box to choose a node for which to lock properties.
Node	Node name.
Result	This field appears after save lock down changes and displays the results of the action.
Lockdown User ID	Select the box to lock down the value defined in the **User ID** field for the selected node.
Lockdown Node Type	Select the box to lock down the value defined in the **Node Type** field for the selected node.
Lockdown Authentication	Select the box to lock down the value defined in the **Authentication Option** field for the selected node.
Lockdown Connectors	Select the box to lock down all property values defined for the node on the Nodes – Connector page in PIA. These properties include the **Connector ID,** the **Gateway ID** and the **Delivery Mode.**
Lockdown WS Security	Select the box to lock down the values defined for the WS Security options defined for the selected node.
Lockdown Active Flag	Select the box to lock down the value defined for the **Active Node** field for the selected node.
Lockdown Node Properties	Select the box to lock down node properties defined for the selected node.
Lockdown Local Default Flag	Select the box to lock down the value defined for the **Default Local Node** field for the selected node
Lockdown Local Flag	Select the box to lock down the value defined for the **Default Local Node** field for the selected node.
Lockdown Segment Aware	Select the box to lock down the value defined for the **Segment Aware** field for the selected node.
Lockdown WSIL URL	Select the box to lock down the value defined for the **WSIL URL** field if the node selected is an external node.
Lockdown Tools Version	Select the box to lock down the value defined for the **Tools Version** field if the selected node is a portal node.

Field or Control	*Description*
Lockdown Application Release	Select the box to lock down the value defined for the **Application Release** field if the selected node is a portal node.

To lock down node properties for project copy:

1. Access the Node Lockdown page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Utilities** > **Lock Down Node Properties.**.)
2. Select the node(s) with which to work.

   To search for a specific node:

   a. In the **Node Name** field, enter the node name or click the **Lookup** button to search for it.

   b. Click the **Search** button.

   To search and display all nodes in the database:

   a. Leave the **Node Name** field empty.

   b. Click the **Search** button.

   The search results appear in the Node List grid.
3. In the **Selected** column, click the box next to each node with which to work.
4. Click the **Primary Lockdown Fields** tab and select the fields to lock.
5. Click the **Secondary Lockdown Fields** tab and select the fields to lock.
6. Click the **Save** button.

   The **Results** field for each node selected displays the results of the actions.

Backing Up the Integration Gateway Properties File

This topic discusses how to:

- Use the Gateway Metadata page.
- Back up the integration gateway properties file.
- Restore a back up copy of the integration gateway properties file.

Understanding Backing Up the Integration Gateway Properties File

The Integration Network features a Gateway Metadata page that enables you to backup the integration gateway properties file (integrationGateway.properties) defined on the local gateway to the database and then later restore the file. This feature is useful for upgrades, enabling you to back up the file before an upgrade, and then restore the file and the associated settings after the upgrade.

To back up the integration gateway properties file or restore a backed up copy of the file, you must enter the gateway user ID and password.

Using the Gateway Metadata Page

To access the Gateway Metadata page (IB_GATEWAYMETA), select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Utilities** > **Back up Gateway Properties.**

This example illustrates the fields and controls on the Gateway Metadata page before you perform a back up. You can find definitions for the fields and controls later on this page.

The information that appears in the top portion of the page reflects the current local gateway ID and the gateway URL that is currently defined in the database. After you perform a backup, the Saved Gateway Properties section of the page displays the gateway URL that was backed up, the publishing node from which the back up was performed, and other data.

This example illustrates the Gateway Metadata page after a back up has been performed. You can find definitions for the fields and controls later on this page.

The following fields and controls appear on this page:

Field or Control	*Description*
Gateway ID	ID of the local gateway.
Gateway URL	The gateway URL that appears at the top of the page is the current URL defined for the local gateway. The gateway URL that appears in the **Saved Gateway Properties** box is the gateway URL defined when the back up was performed.
Gateway Setup Properties	Click the link to access the PeopleSoft Node Configuration page and the integration gateway properties file to view current settings or make modifications.
Publisher	Name of the node from which the back up or restore was performed.
Last Update Data/Time	Date and time that the last action, back up or restore, was performed.
Show Saved Gateway Properties	Click the link to view the integration gateway properties file that was backed up. This link is enabled only after a back up is performed.
Backup Gateway Properties	Click the button to save a copy of the integration gateway properties file to the local database.
Restore Gateway Properties	Click the button to restore the integration gateway properties file.

Backing Up the Integration Gateway Properties File

To back up the integration gateway properties file:

1. Access the Gateway Metadata page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Utilities** > **Back up Gateway Properties.**).
2. Click the **Backup Gateway Properties** button.

   The Gateway Properties page appears.
3. Enter the integration gateway security credentials:

   a. In the **User ID** field, enter the gateway user ID.

b. In the **Password** field, enter the gateway password.

c. Click the **OK** button.

A message appears indicating that the action was successful and the Gateway Metadata page appears. The **Show Saved Gateway Properties** link appears. Click the link to view the backed up file.

Restoring a Backed Up Copy of the Integration Gateway Properties File

To restore a backed up copy of the integration gateway properties file:

1. Access the Gateway Metadata page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Utilities** > **Back up Gateway Properties.**).
2. Click the **Restore Gateway Properties** button.

   The Gateway Properties page appears.
3. Enter the integration gateway security credentials:

   a. In the **User ID** field, enter the gateway user ID.

   b. In the **Password** field, enter the gateway password.

   c. Click the **OK** button.

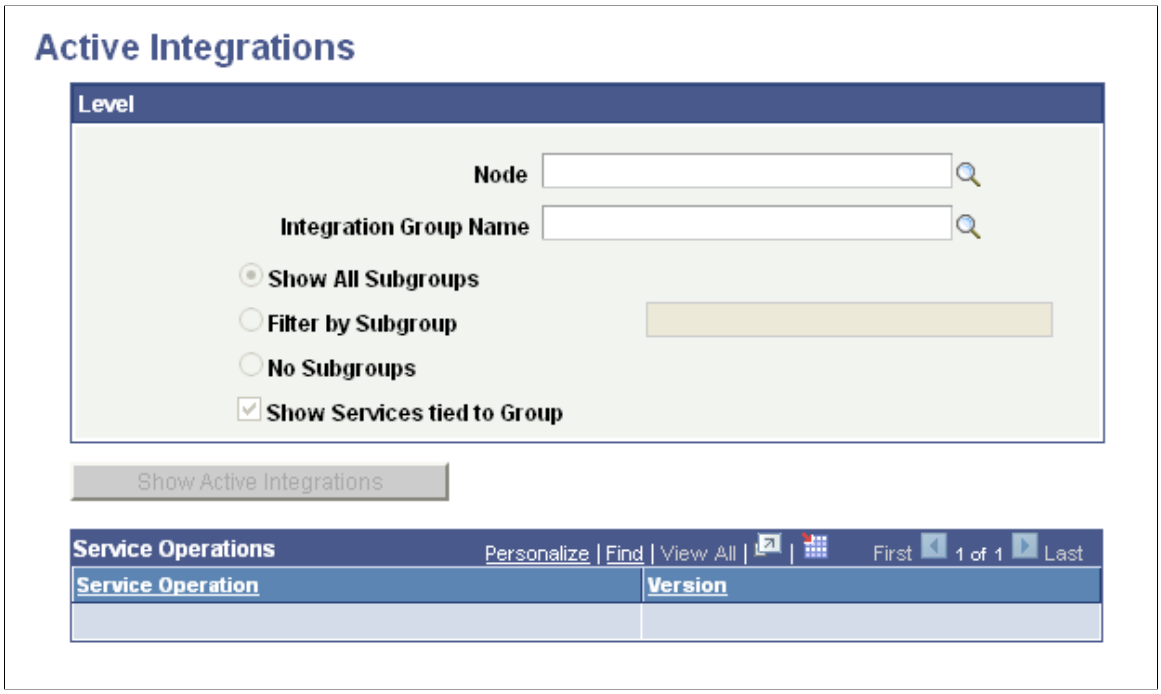
A message appears indicating that the action was successful and the Gateway Metadata page appears.

Viewing Active Integrations

Use the Active Integrations page (IB_ACTIVE_SERVICES) to view active integrations in the database.

To access the Active Integrations page, select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Utilities** > **View Active Integrations.**

This example illustrates the fields and controls on the Active Integrations page. You can find definitions for the fields and controls later on this page.



You must search by node and integration group name. The subgroup search criteria are optional.

After you enter your search criteria, the **Show Active Integrations** button becomes enabled. Click the button to search the database for active integrations based on the search criteria entered. The results of the search appear in the Service Operations results grid at the bottom of the page.

The page features the following fields and controls:

Field or Control	***Description***
Node	Enter the node name.
Integration Groups Name	Enter the integration group name.
Show All Subgroups	Select the control to list all service subgroups. (Default.) The search results show the name of the service only.
Filter by Subgroup	Select the control to list only the services for a specific subgroup. Enter the name of the subgroup by which to filter in the field provided. The search results show all service operations for the service subgroup specified.

Field or Control	*Description*
No Subgroups	Select the control to exclude from the search results any service subgroups defined for the integration group.
Show Services tied to a Group	Select the box to show services associated with the integration group.
Show Active Integrations	Click the button to search the database for active integrations based on the criteria entered.
Service Operation	Displays the service operation name.
Version	Displays the service operation version.

Chapter 12

Using the Integration Hub

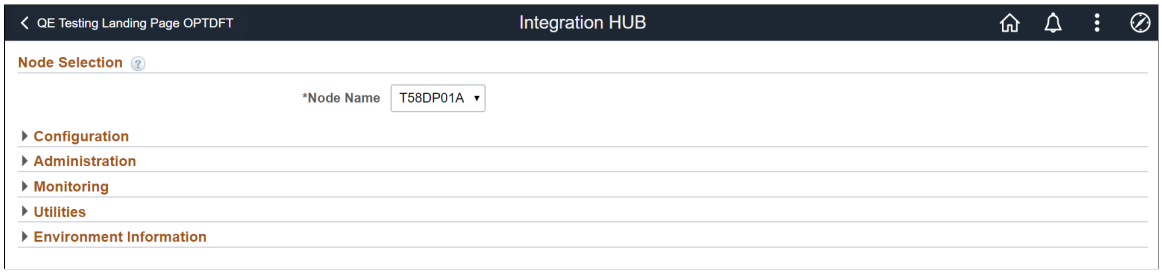
Understanding the Integration Hub

The Integration Hub centralizes the administrative tasks of configuring, administering, and monitoring integrations in PeopleSoft Integration Broker and allows the administrator to perform tasks on all applicable pillars (nodes in the network) with a single sign on.

Accessing the Integration Hub

To access the Integration Hub select **PeopleTools** > **Integration Broker** > **Integration Hub.**

This example illustrates the default view of the Integration Hub.



The Integration Hub provides the following options for configuring, administering, and monitoring integrations in PeopleSoft Integration Broker:

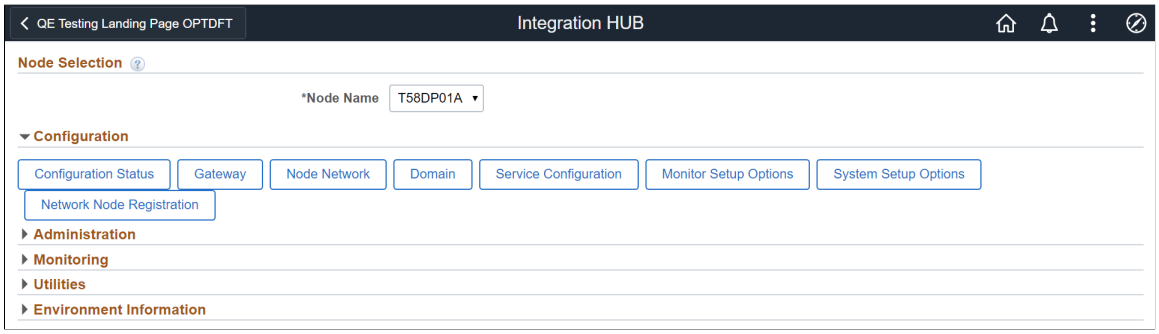
Field or Control	***Description***
Node Name	Allows you to select a node (pillar) from a list of all the nodes defined in the network.
Configuration	Displays links which guide the administrator through the Integration Broker Setup process.
Administration	Displays links which allow the administrator to enable and check the status of integrations.
Monitoring	Displays links which allow the administrator to monitor integrations across the defined network.
Utilities	Performs various functions such as activating integrations, displaying certain environment information and so on.

Field or Control	*Description*
Env Information	Displays links which allow the administrator to look at various configuration settings.

Configuration

Click the Expand button for Configuration to access the links under it.

The following image displays the links available under Configuration.

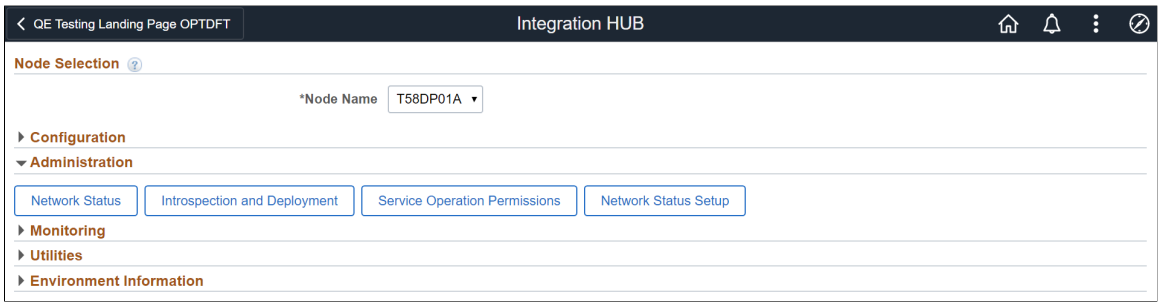


Field or Control	*Description*
Configuration Status	Shows the configuration status of the domain (Gateway, Node Network, and Domain).
Gateway	Configures the gateway with local and remote nodes.
Node Network	Selects and configures nodes for the network.
Domain	Registers remote nodes defined on the integration gateway with the PeopleSoft Integration Network.
Service Configuration	Configures server, schema namespaces, and target locations.
Monitor Setup Options	Enables gateway logging and data view size limit.
System Setup Options	Sets maximum recursion level and enables IB Profiling.
Network Node Registration	Activates the domain for asynchronous transaction processing.

Administration

Click the Expand button for Administration to access the links under it.

The following image displays the links available under Administration.

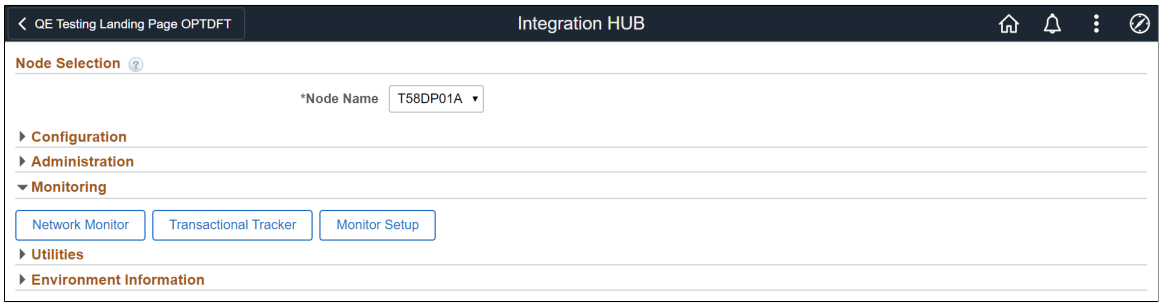


Field or Control	***Description***
Network Status	Performs Network Status checks or IB pings to all selected network nodes.
Introspection and Deployment	Introspects and deploys all Service Operations to selected network nodes allowing the administrator to enable and/or create the appropriate routing(s).
Service Operation Permissions	Add/ Modify Service Operation Permissions to all local defined Service Operations.
Network Status Setup	Setup Information used for data retrieval, automated integration status and diagnostics.

Monitoring

Click the Expand button for Monitoring to access the links under it.

The following image displays the links available under Monitoring.



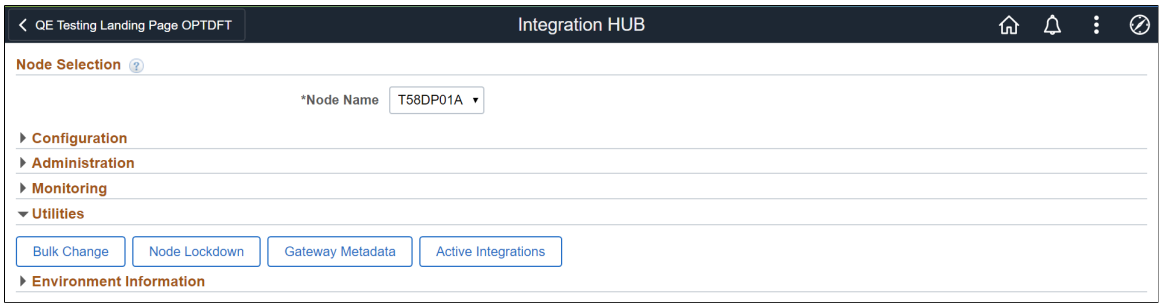
Field or Control	***Description***
Network Monitor	Monitors the remote Service Operation Monitor.

Field or Control	*Description*
Transactional Tracker	Tracks specific network transactions between local and selected network nodes.
Monitor Setup	Sets up query information for data retrieval from network nodes.

Utilities

Click the Expand button for Utilities to access the links under it.

The following image displays the links available under Utilities.

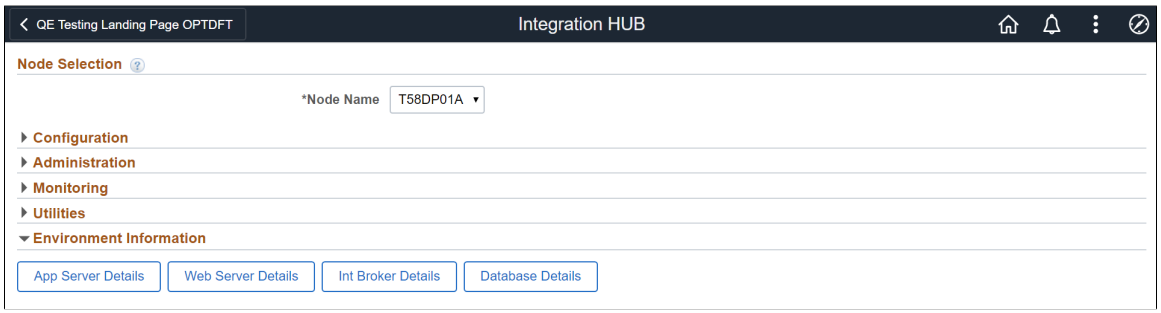


Field or Control	*Description*
Bulk Change	Enables you to change either node or routing information, or both, in bulk.
Node Lockdown	Locks down properties of the node such that a Project Copy does not override the node metadata.
Gateway Metadata	Contains a backup Integration.Properties gateway file.
Active Integrations	Activates all service metadata.

Environment Information

Click the Expand button for Environment Information to access the links under it.

The following image displays the links available under Environment Information.



Field or Control	***Description***
App Server Details	Activates all service metadata.
Web Server Details	Displays web server configuration information.
Int Broker Details	Displays Integration Broker configuration information.
Database Details	Displays database configuration information.

Related Links

Understanding the Integration Hub
Configuring the Integration Hub

Configuring the Integration Hub

To configure and enable the Integration HUB link:

1. For the nodes that will be defined in the network, populate the Check Token ID.
2. Ensure that the default local node and the Portal node(s) on each pillar defined in the network include the proper URIs on the Portal page for the Context URI Text and the Portal URI Text.

   Warning! The URL(s) that are used (either http or https) needs to be consistent to how the user is logged in to invoke the application. For example, if the default local node and Portal node(s) URLs defined on the Portal page for Context URI Text and the Portal URI Text are secure URLs (i.e. https) for ALL nodes in the network, then the user needs to log in via https. Mismatch of secure and non-secure URLs will result in the page selected not properly rendering.

3. Ensure that Single Signon is properly enabled for all nodes. This requires that each remote node defined in the Integration Network on each pillar has a trust authentication token issued by the Node. This will allow the PSTOKEN generated from Single Signon to be used as validation on any of the other nodes in the Integration Network. See “Implementing PeopleSoft-Only Single Signon” (Security Administration).

4. Enable "Allow Domain Compare" on the Authorized Site page for the applicable Web Profile (People Tools\ Web Profile\Web Profile Configuration).

5. Perform a successful network ping from each pillar that is defined in the network (or at least where the Integration HUB will be used). See Verifying Integration Processing in the Integration Network.

6. For the end user to have access to the Integration Hub page, ensure that the user(s) are added to the following permission lists - PTPT1200 and PTPT1000. See Setting Service Operation Permissions.

Chapter 13

Activity Guide: Configuring PeopleSoft Integration Broker

Understanding the Integration Broker Configuration Activity Guide

This topic provides overview information about the Integration Broker Configuration activity guide.

Understanding the Integration Broker Configuration Activity Guide

The PeopleSoft Integration Broker Configuration activity guide is a pagelet in the Integration Network WorkCenter. The activity guide provides centralized access to the PeopleSoft Pure Internet Architecture (PIA) pages used to configure PeopleSoft Integration Broker and the Integration Network.

The purpose of the PeopleSoft Integration Broker Configuration activity guide is to lead users through a basic set up of PeopleSoft Integration Broker and an integration network. This activity guide is intended for users who need Integration Broker configured and running to use other PeopleTools technologies, such as the Feeds framework, the PeopleSoft Search framework, the PeopleSoft Test framework, Unified Navigation, and others. It is also intended for users who are learning Integration Broker or those who need to a basic Integration Broker configuration for a sandbox environment.

Note: Additional configuration may be required based on the PeopleTools technology you are using. Please refer to the product documentation for specific technologies for information about additional configuration requirements.
Integration developers, integration administrators and super users will likely have additional configuration requirements and should use the Integration Network or the traditional PIA pages for configuring PeopleSoft Integration Broker.

The goal of this topic is to provide information that enables you to get Integration Broker up and running. The documentation in this section does not provide extensive detail on why a configuration activity is required and does not discuss advanced configuration options. For additional information about a configuration step and additional configuration options, please refer to the online Help available on each PIA page and the PeopleTools documentation library.

Understanding the Integration Broker Configuration Activities

The following table describes the activities in the Integration Broker Configuration activity guide

Step	Activity	Description
1	Set up gateway.	In this activity you: • Define the gateway URL and load target connectors. • Register nodes on the local gateway. • Define integration gateway keystore values. **Important!** You must define an encrypted keystore password during gateway setup. If you do not enter an encrypted password integration will fail.
2	Add target locations.	In this activity you: • Define the schema namespace and the service namespace. • Define the target location.
3	Register nodes with the network.	In this activity you: • Set the network node password. • Confirm that integration gateway keystore values are set. • Register nodes with the network.
4	Activate domain.	In this activity you activate the application server domain.
5	Check network connections.	In this activity you verify that the local node can communicate with other PeopleSoft nodes defined in the integration network.
6	Introspect and deploy integrations.	(Optional.) In this activity you introspect and deploy integrations among the integration partners defined in the network. This activity may not be required in all cases.
7	Update security on service operation permissions.	In this activity you assign permission lists and access levels to service operations.

Prerequisites for Configuring Integration Broker Using the Activity Guide

The following prerequisites must be met to configure Integration Broker using the activity guide:

- Install a database, web server, PeopleTools and PIA.
- Configure and boot an application server domain.
- Add and configure node definitions.

- Define a target connector.

Installing PeopleTols, Database, Application Server, Web Server (PIA), and Process Scheduler

You need a database, web server, PeopleTools, and PIA installed before beginning the activities in this guide.

See *PeopleSoft 9.2 Application Installation* for your database platform.

During the PeopleSoft DPK deployment:

- You are prompted to set the integration gateway user ID and password. Note these values as you will use them frequently to configure and manage the integration gateway.
- Note the values for the Jolt port that you define. You will use this value to configure the application server domain and also to configure aspects of the integration system.
- Note the value for the HTTP or HTTPs port that you define. You will use this value to configure aspects of the integration system.

The default values for these ports are defined in one of the files used by the DPK process, psft_configuration.yaml.

To define other ports for the installation, see "Completing the DPK Initialization with Customizations" in the installation documentation.

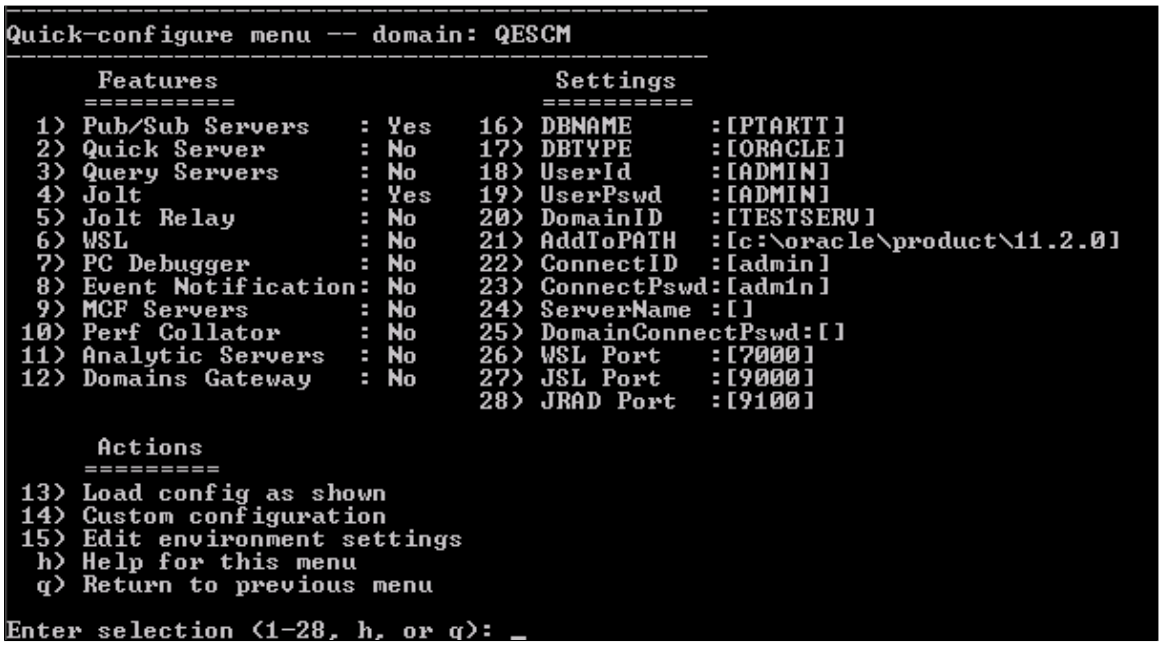
See *PeopleSoft 9.2 Application Installation* for your database platform.

Configure and Boot an Application Server Domain

You must have an application server created, configured and booted before beginning the activities in this guide.

Ensure that the Feature option **1.) Pub/Sub Servers** is set to *Yes.* The PeopleSoft DPK deployment sets this by default. Use PSADMIN to verify the setting. This setting is required to perform asynchronous integrations. This setting is also required for the activity of activating the application server domain described later in the activity guide.

This example illustrates the Quick Configure Menu in PSADMIN.



The settings shown in the Feature column in the previous example are adequate for a basic configuration or a sandbox environment. Note that the values for the options in the Settings column are dependent on the specifics of the database installation, PeopleTools installation, PIA installation, settings in Configuration Manager, and so on.

Adding and Configuring Node Definitions

Each database involved in an integration must contain a default local node definition for itself and a remote node definition for each of the other nodes (integration partner systems) involved. The Understanding Nodes section of the documentation provides a subsection on local and remote nodes that you may find helpful if you are not familiar working with the concepts of local and remote nodes.

See Understanding Nodes

Although the procedures for adding and configuring nodes is described in the product documentation, the following information will help to reduce the time to add and configure node definitions for use in the integration network.

- When you configure a node, you define a node type. To use the features of the integration network, the local and the remote nodes must be defined as *PIA* node types. There is a **Node Type** option on the node definition page where you set the value. By default, the default local node is a *PIA* node type.

- When you define a node, you specify a node password and a default user ID. Provide this information to each of your integration partners participating in the network. They must define this information in the remote node definition that they create in their databases to represent your system.

  In turn, each of your integration partners must provide you with the password and user ID used on their local default node definition so that you can define this information in the remote node definitions that you create for each of their systems in your local database.

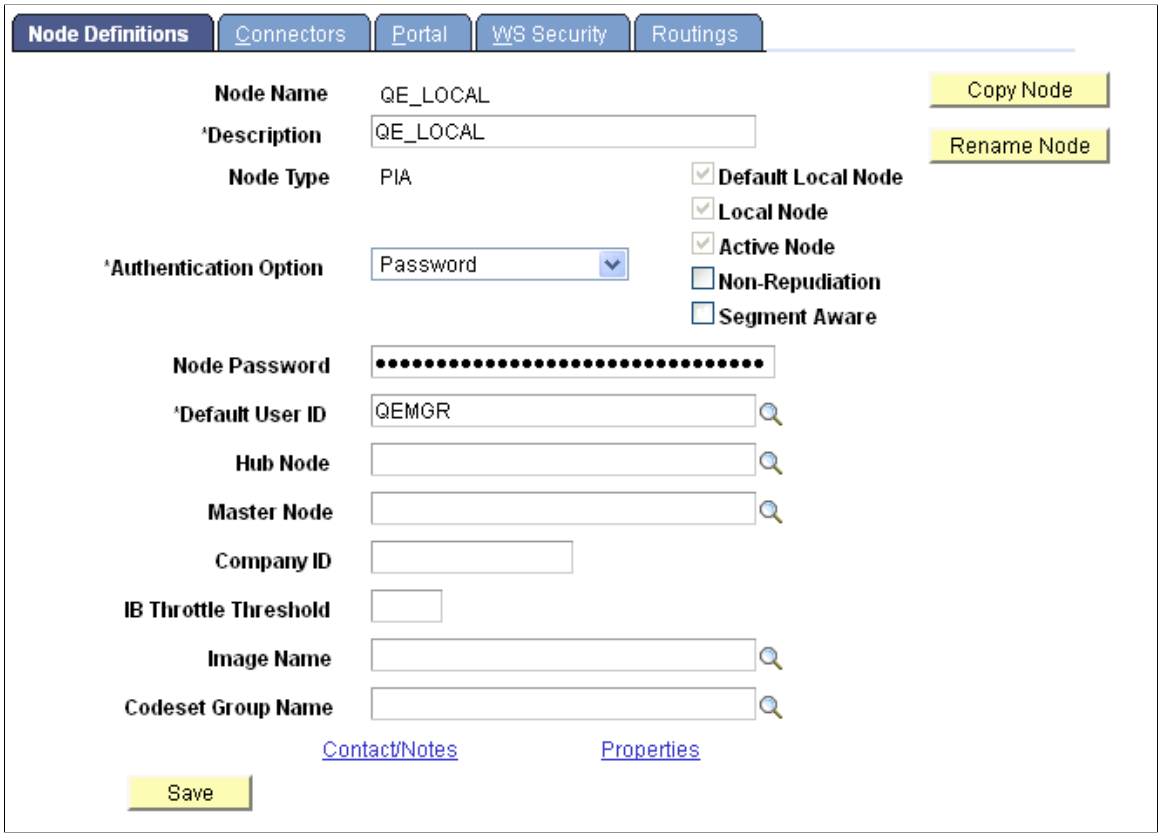
Note that you may have to create in your database the user profile for the user ID that your integration partners give you for their local node, and vice versa.

- All nodes definitions must be active. There is an **Active** control box in the node definition that you must select.
- PeopleTools delivers a default local node definition, *QE_LOCAL*. For a basic configuration you can use this definition out-of-the-box with little additional configuration required. The node is delivered as an active, local , PIA node. You may need to change the password or the user ID.

  Some PeopleTools technologies and product areas may deliver nodes for use with the specific technology. If you are configuring Integration Broker as a prerequisite for using another PeopleTools technology or product, consult the product documentation to learn if nodes are delivered for use with the technology and to discover if any additional node configuration requirements exist..

Use the Nodes – Node Definitions page to add and configure node definitions. To access the page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions**.

This example illustrates a node definition for the default local node.

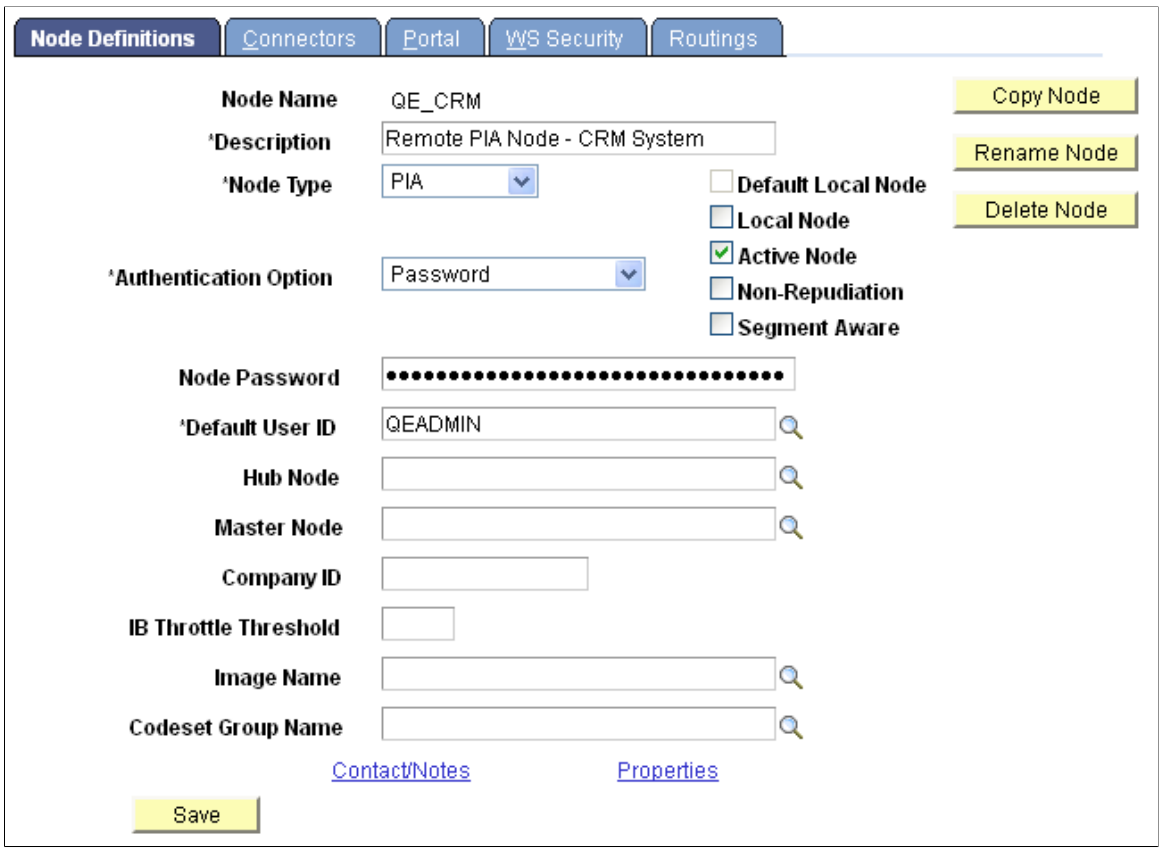


This example shows the default local node definition, *QE_LOCAL*.

Each of your integration partners must create a node definition on their local database that exactly matches your default node definition. You provide your partners with the node name, node type, authentication option, node password, and default user ID defined. They then create a node definition in their local database using the information. The only differences between the node definition in your database and the definition that your integration partners create are that your integration partners leave clear the default local node and local node fields.

You must create a node definition on the local database to represent each of your integration partner system.

This example illustrates a remote node definition for an integration partner on the local database.



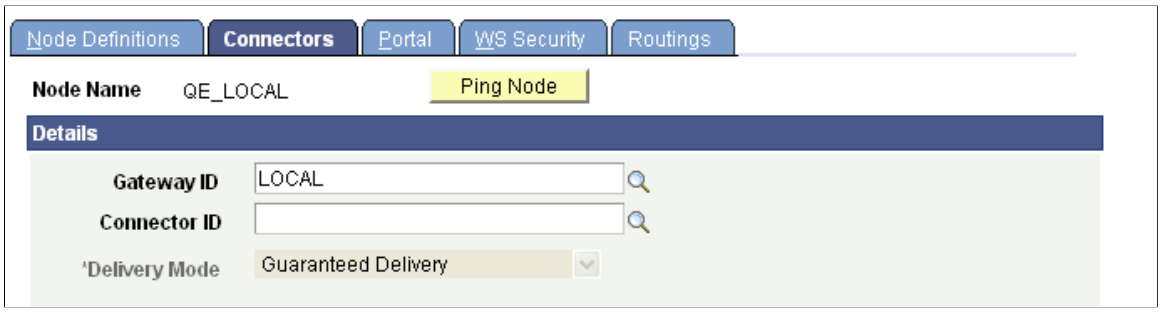
The previous example shows a node definition in the local database for an integration partner. Remote node definitions must match the definition in your integration partner's local database. Based on the example, the integration partner would have communicated to you the node name in their local database, the node type, the authentication option, the node password, and the default password for the definition. Note that the default local node and local node fields are not selected, since this is a remote node definition. Assuming that this definition matched the definition on your integration partner's local database, this definition is an adequate node definition.

Defining a Target Connector

In the PeopleSoft Integration Broker framework, target connectors generate requests, send them to integration participants, wait for responses from participants, and deliver the responses back to the gateway.

Define target connectors on the Connectors page in the node definition. To access this page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** and click the Connectors tab.

This example illustrates the Nodes – Connectors page.



PeopleSoft delivers a PeopleSoft target connector for use with integrations among PeopleSoft systems. Enter or select *PSFTTARGET* from the list to select the connector.

If you are configuring the integration system to use another PeopleTools technology or product, consult your product documentation to determine if you need to use a different target connector.

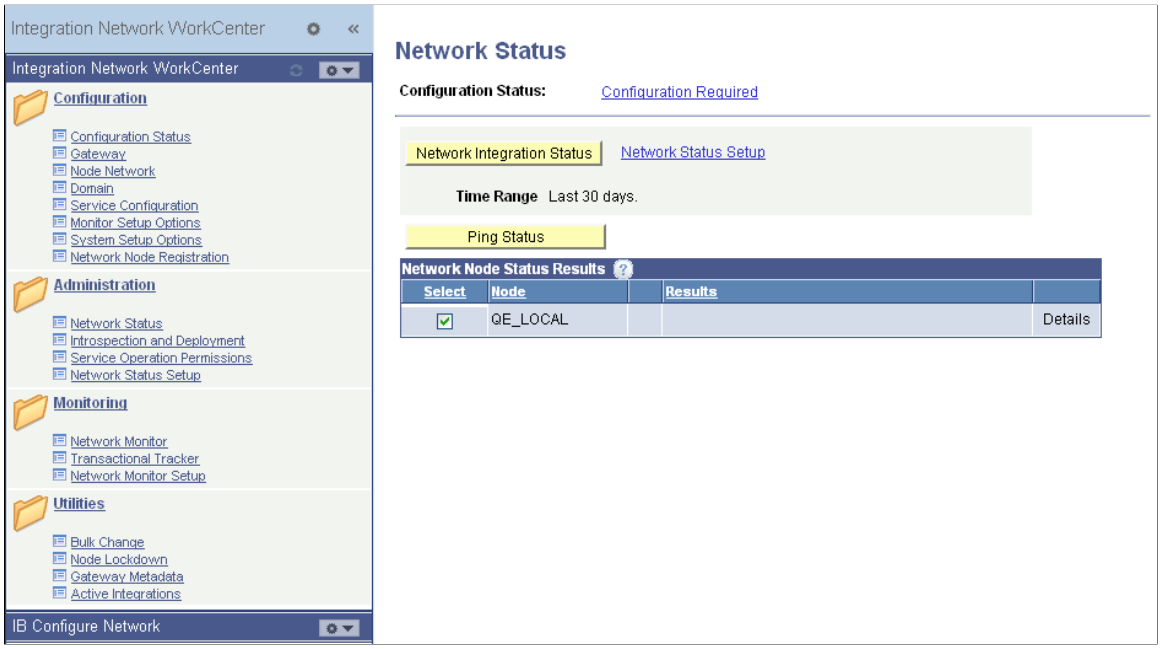
You define target connectors on local and remote node definitions. The target connector that you define on a remote node definition must match the target connector that is defined on your integration partner's local node definition. Likewise, the target connector that your integration partner defines on the remote node definition for your system must match the connector defined on your local node definition.

Accessing and Navigating the Integration Broker Configuration Activity Guide

To access the Integration Broker Configuration activity guide select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter.**

By default, the Integration Network pagelet appears in the work area.

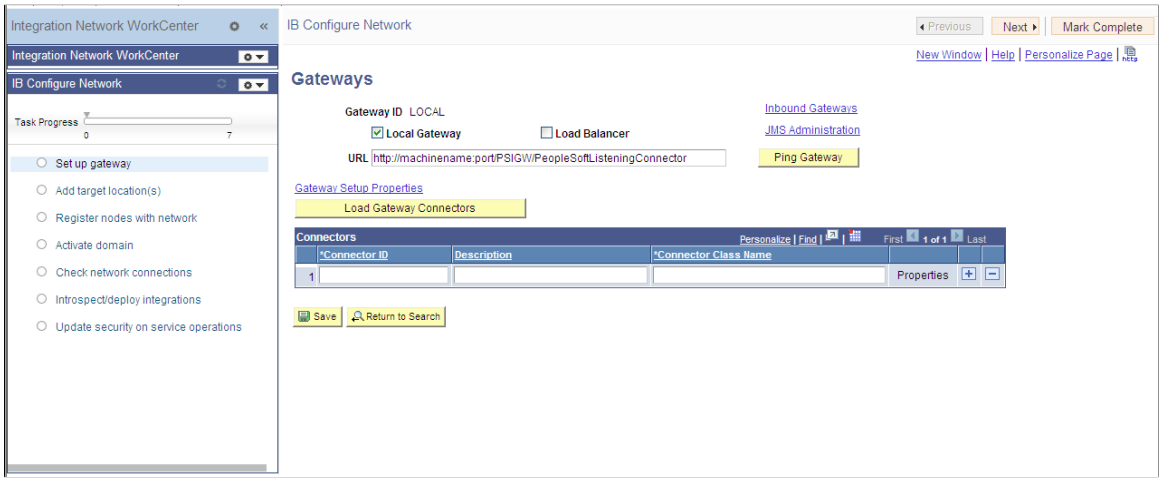
This example illustrates the default view of the Integration Network WorkCenter. The Integration Network WorkCenter pagelet is expanded and the Integration Broker Configuration pagelet is minimized at the bottom of the workcenter.



To access the Integration Broker Configuration activity guide, minimize the Integration Network WorkCenter pagelet and maximize the Integration Broker Configuration pagelet.

Field or Control	***Description***
	This icon is an expand/minimize toggle. Click icon in the upper right corner of the WorkCenter pagelet window to expand or minimize the pagelet.

This example illustration the default view of the Integration Broker Configuration activity guide. The fields and controls that appear on all pages in the activity guide are described later on in this topic.



Use the following fields and controls to navigate in the activity guide:

Field or Control	*Description*
	Click the icon to refresh the configuration status of activities in the guide.
Task Progress	This control illustrates the progress in completing the tasks in the activity guide. The on the far right under the task bar indicates the total number of tasks in the guide. As you complete an activity, the progress bar moves forward.
Previous	Click the button to go back to the previous activity.
Next	Click the button to go to the next activity.
Mark Complete	Click the button to mark an activity complete. When you mark an activity complete, the circular icon next to an activity becomes solid green.

The activities in the guide are listed under the task bar. Click an activity name to access the corresponding pages to complete the activity.

An activity can have the following states:

Field or Control	*Description*
	A light gray circle icon that appears next to a task indicates that the task has not been visited or started.
	A solid green circle icon that appears next to a task indicates that the task has been visited or is in progress, but it is not complete.
	A check icon that appears next to a task means that the task is complete.

Activity 1: Setting Up the Integration Gateway

In this activity you:

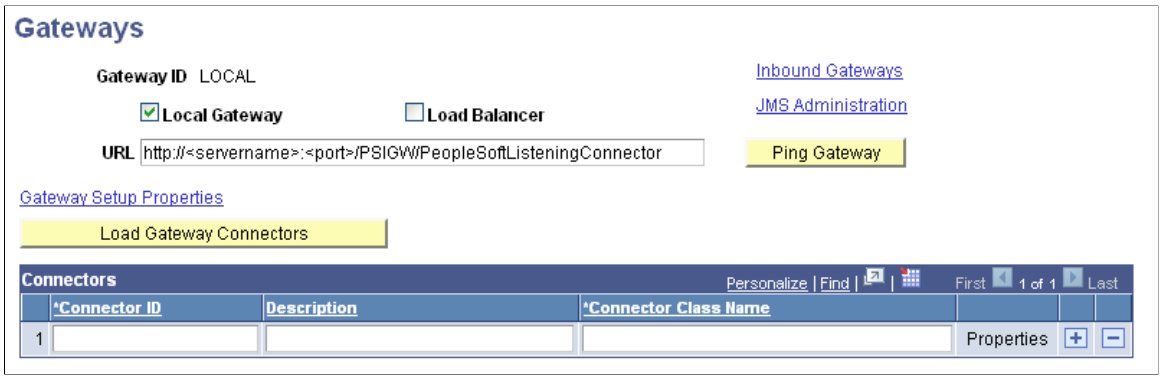
- Define the integration gateway URL and load target connectors.
- Register nodes on the local gateway.
- Define integration gateway keystore values

Defining the Integration Gateway URL and Loading Target Connectors

The integration gateway URL specifies the location of the PeopleSoft listening connector used to “listen” for inbound request/integrations. The target connectors handle outbound requests/integrations.

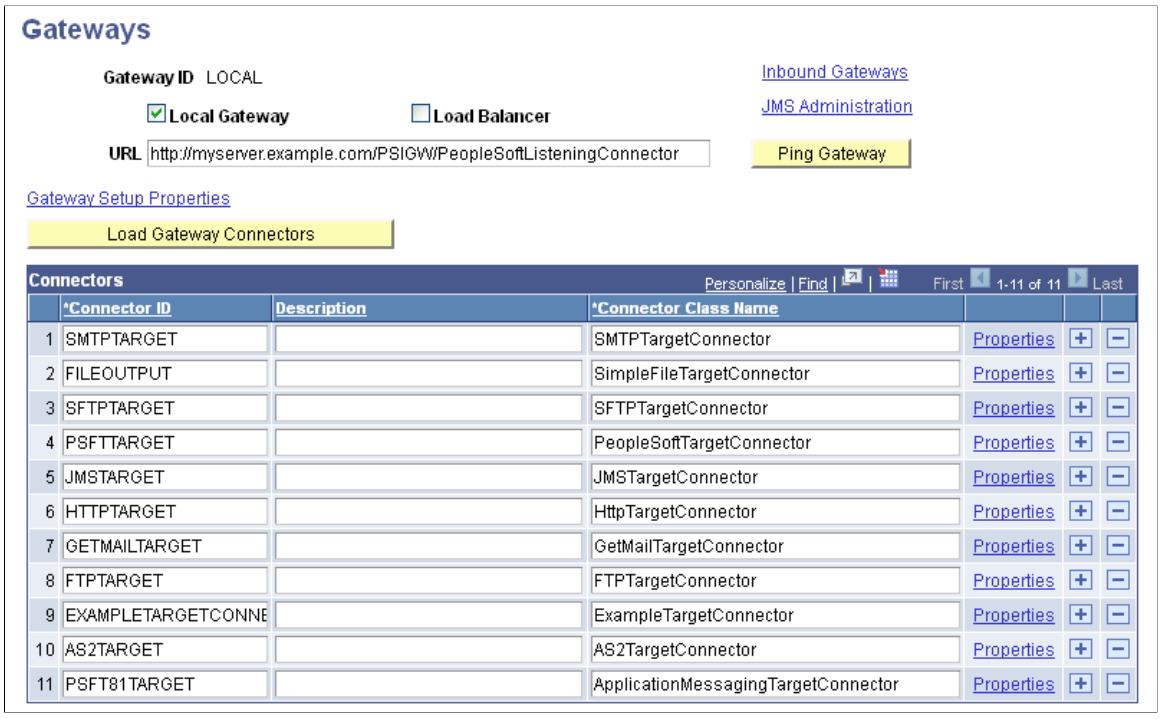
To define the integration gateway URL and load target connectors use the Gateways page. To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Integration Broker Configuration** > **Set Up Gateway.**

This example illustrates the Gateways page as it appears prior to any configuration.



After you complete the configuration tasks on this page the **URL** field is populated with the URL where PIA is installed and the target connectors delivered with PeopleSoft Integration Broker are populated in the Connectors grid.

This page illustrates the Gateways page when the gateway URL is defined and the target connectors are loaded.



You can test proper connectivity to the gateway by using the **Ping Gateway** button. The results of the ping appear in a new window.

This example illustrates the PeopleSoft Integration Gateway page and the results of pinging the integration gateway after configuring the gateway URL and loading the target connectors. The status shown in the example is *ACTIVE* and indicates proper communication with the gateway.

To define the gateway URL and load target connectors:

1. Access the Integration Broker Configuration activity guide (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** and expand the Integration Broker Configuration pagelet).
2. In the left navigation page, click **Set Up Gateway.**

   The Gateways page appears.
3. In the URL field enter the gateway URL in the following format:

   `http://<machinename>:<port>/PSIGW/PeopleSoftListeningConnector`

   The machine name is the name of the machine where PIA is installed. By default the port number is *80* for HTTP and *443* for HTTPS. If using the default port number, you do not need to specify it in the URL.

   An example URL, assuming the default HTTP port is used, is shown in the following example:

   `http://myserver.example.com/PSIGW/PeopleSoftListeningConnector`
4. Click the **Load Gateway Connectors** button.

   A message appears that indicates the connectors were successfully loaded.
5. Click the **OK** button.
6. Click the **Save** button.
7. Click the **Ping Gateway** button.

   A successful ping means that the system can properly communicate with the gateway.

   If the ping is not successful, check that the correct URL is entered and that it is entered in the proper format..

Important! When you save the changes on the Gateways page, the system marks the “Setting up gateway” activity complete. However you still need to register nodes on the local gateway and define integration gateway keystore values for the gateway configuration to be complete.

Registering Nodes on the Local Gateway

You must register the local node and any PeopleSoft nodes on the integration gateway. If you are integrating with other PeopleSoft systems, you must register their associated nodes on the local gateway. In turn, your integration partners must add a node definition in their databases that represents your system and register your node on their gateways.

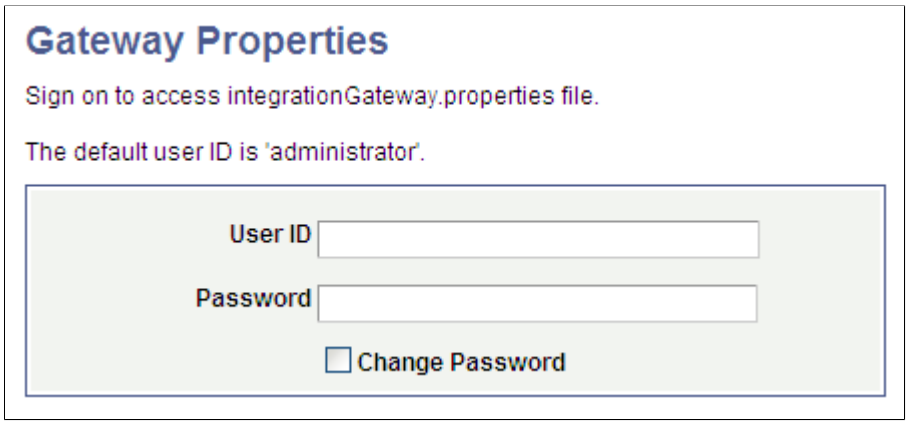
You must also provide an application URL that specifies the application server to process integrations if no valid target node to process the integration can be determined.

The technology that you are using may require that you register other PeopleSoft nodes on the gateway. See the product documentation for the technology that you are using for more information.

Use the PeopleSoft Node Configuration page to register nodes on the gateway. This is a password-protected page that you access from the Gateways page. To access the page use the user ID and password that you defined for the integration gateway during the installation of PIA.

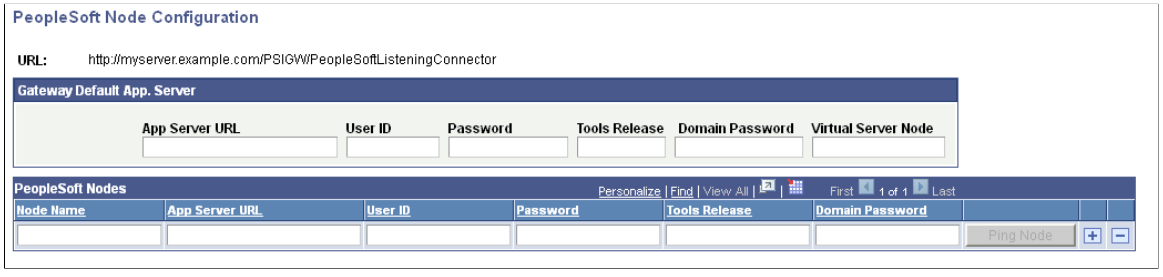
To access PeopleSoft Node Configuration page, from the Gateways page, click the Gateway Setup Properties link. The Gateway Properties page appears and you must enter the gateway user ID and password that were set up during the PIA installation process.

This example illustrates the Gateway Properties page.



After you successfully enter the integration gateway security credentials, the PeopleSoft Node Configuration page appears.

This example illustrates the PeopleSoft Node Configuration page prior to any configuration.



The Gateway Default App. Server section is where you define the application server to process integrations if inbound integrations do not specify a target node. The PeopleSoft Nodes section is where you register PeopleSoft nodes that will handle integrations.

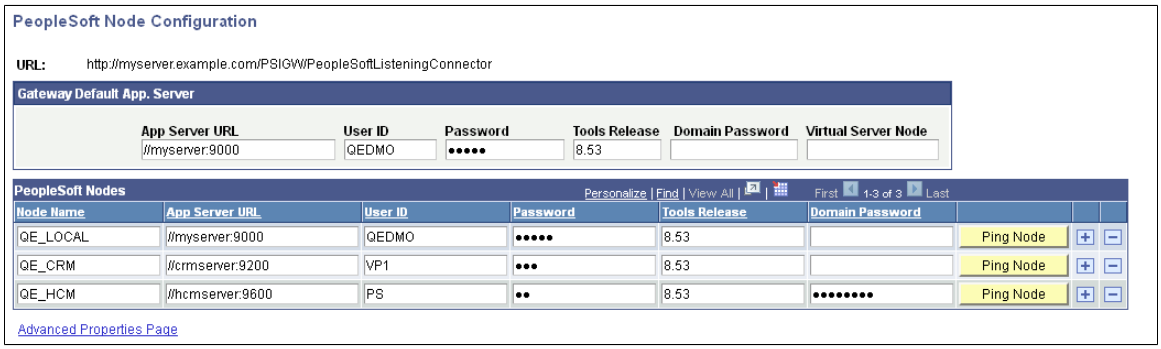
To register nodes for your PeopleSoft integration partners, you must obtain the following information from them:

- Node name.

  The node should be defined in the database.

- Application server URL.
- Application server user ID.
- Application server password.
- PeopleTools release number.
- Application server domain password (if applicable).

This example illustrates the PeopleSoft Node Configuration page. The Gateway Default App. Server section shows that an application server has been registered to handle inbound integrations that do not specify a target node. The PeopleSoft Nodes grid shows the default local node, *QE_LOCAL,* as well as several other PeopleSoft nodes registered.



To register nodes on the local gateway:

1. On the Gateways page, click the **Gateway Setup Properties** link.

   The Gateway Properties page appears.

2. Enter the user ID and password for the integration gateway.

   These credentials were defined when PIA was installed.

   a. In the **User ID** field, enter the gateway user ID.

   b. In the **Password** field, enter the gateway password.

   c. Click the **OK** button.

      The PeopleSoft Node Configuration page appears.

3. In the Gateway Default App Server grid, define the following:

   - In the **App Server URL** field enter the machine name and Jolt port of the application server.

     The format is *//<machinename>:<port>*.

An example is *//myserver:9000.*

Note: The App Server URL is case sensitive in Windows.

- In the **User ID** field, enter the application server user ID.
- In the **Password** field, enter the application server password.
- In the **Tools Release** field, enter the release number of the PeopleTools version installed. For example, *8.55.*
- In the **Domain Password** field, enter the domain password if one was defined when the application server was configured.

4. In the PeopleSoft Nodes grid, define the following fields for the local default node.

Note: Depending on the technology you're using, you may need to define other PeopleSoft nodes in the grid

- In the **Node Name** field, enter the name of the local default node.
- In the **App Server URL** field enter the machine name and Jolt port of the application server.

  The format is *//<machinename>:<port>*.

  An example is *//myserver:9000.*
- In the **User ID** field, enter the application server user ID.
- In the **Password** field, enter the application server password.
- In the **Tools Release** field, enter the release number of the PeopleTools version installed. For example, *8.55.*
- In the **Domain Password** field, enter the domain password if one was defined when the application server was configured.

5. Click the **Ping Node** button.

   A successful ping means that the integration gateway can communicate with the node.

Define Integration Gateway Keystore Values

You must define an encrypted keystore password in the integration gateway properties file. The path to the keystore is populated during the PIA installation process, but it's good practice to confirm that the path is accurate during this task.

Important! Integrations will fail if you do not enter an encrypted keystore password for the secureFileKeystorePasswd property.

You define the keystore values in the

The following code snipped shows an example of the keystore values before they are configured:

```
secureFileKeystorePath=C:/Users/admin/psft/pt/8.55/webserv/peoplesoft/piaconfig/key⇒

store/pskey
#secureFileKeystorePasswd=
```

To configure these settings confirm that the path to the keystore is correct. Next, encrypt the keystore password with the provide encryption utility, uncomment the secureKeystorePasswd property, and set the property equal to the encrypted password.

The following code snippet shows an example of the keystore values after they are properly configured:

```
secureFileKeystorePath=C:/Users/admin/psft/pt/8.55/webserv/peoplesoft/piaconfig/key⇒

store/pskey
secureFileKeystorePasswd={V1.1}7m4OtVwMGDyLc1j6pZG69Q==
```

To define integration gateway keystore values:

1. From the PeopleSoft Node Configuration page, click the **Advanced Properties Page** link.

   The Gateway Properties page appears.

2. Scroll to the ## Integration Gateway CERTIFICATE Section of the file.

3. Locate the secureFileKeystorePath property.

4. Confirm the keystore path setting:

   a. Uncomment the secureFileKeystorePath property if it is not already uncommented.

   b. Confirm that the path defined for the keystore path is accurate.

5. Enter an encrypted keystore password.

   a. Expand the Password Encryption utility at the bottom of the page.

   b. In the **Password** field, enter the keystore password.

   c. In the **Confirm Password** field, enter the password again.

   d. Click the **Enter** button to encrypt the password.

      The encrypted password appears in the **Encrypted Password** field.

   e. Copy the value in the **Encrypted Password** field to the clipboard.

   f. Navigate back to the secureFileKeystorePasswd property in the file.

   g. Uncomment the property and paste the encrypted value, setting the property equal to the encrypted value.

6. Click the **OK** button

Related Links

Administering Integration Gateways

Activity 2: Adding Target Locations

In this activity you:

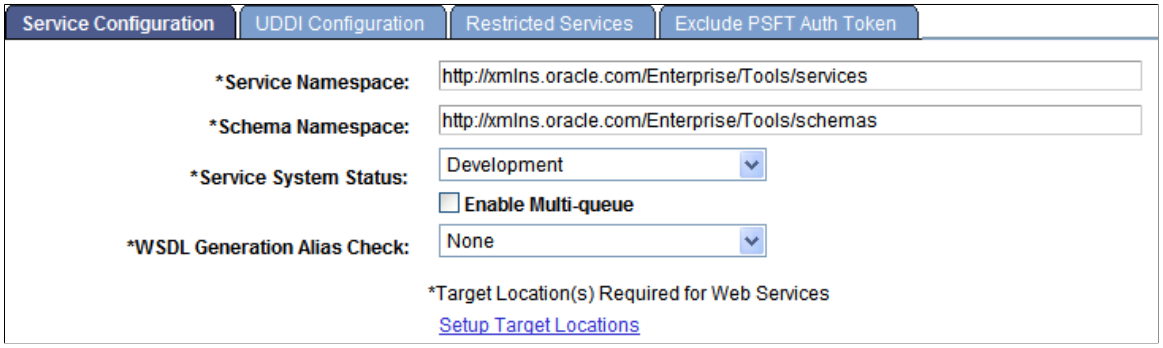
- Define the schema namespace and the service namespace.
- Define target locations.
- Define the service system status.

Defining the Schema Namespace and the Service Namespace

Namespaces provide a method for qualifying element and attribute names that are used in XML documents and are identified by Uniform Resource Identifier (URI) references.

To define the schema namespace and the service namespace, use the Service Configuration page. To access the page **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network** and click the **Add Target Location(s)** link in the left navigation pane.

This example illustrates the Service Configuration page.



PeopleTools provides the following default namespaces:

Field or Control	***Description***
Service Namespace	`http://xmlns.oracle.com/Enterprise/Tools/services`
Schema Namespace	`http://xmlns.oracle.com/Enterprise/Tools/schemas`

You can use the default values or define different values.

To define the schema namespace and the service namespace:

1. Access the Integration Broker Configuration activity guide (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network**).
2. In the left navigation pane click **Add Target Location(s).**

   The Service Configuration page appears.
3. In the **Service Namespace** field, enter the service namespace.
4. In the **Schema Namespace** field, enter the schema namespace.
5. Click the **Save** button.

Defining Target Locations

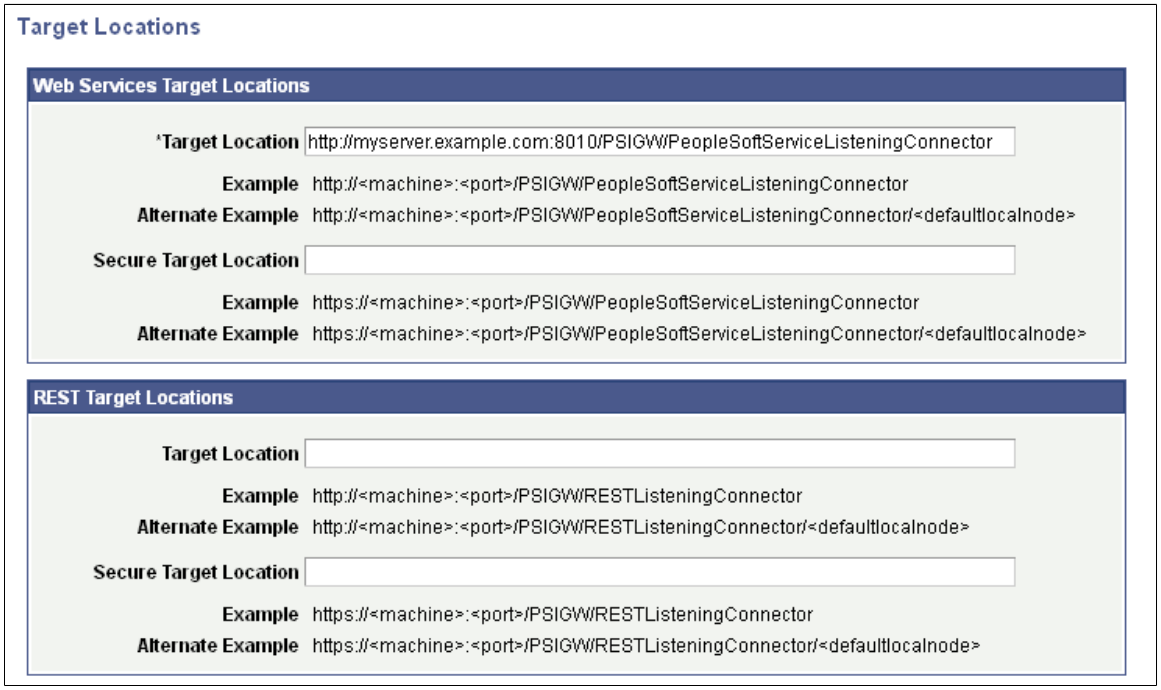
Target locations are URLs that PeopleSoft Integration Broker uses to build and validate XML message schemas, export WSDL documents, and as the SOAP endpoint. For REST services, target locations are URLs that PeopleSoft Integration Broker uses to export WADL documents and as the REST endpoint.

In general, the URL you specify as a target location should be an unsecured URL. If you need to enter secure target URLs, please see the product documentation for additional information before proceeding.

You need to set the REST target location only if performing integrations with REST-based services.

Use the Target Locations page to define target locations. To access the page, click the **Set Target Locations** link on the Service Configuration page.

This example illustrates the Target Locations page.



The Target Locations page provides examples of the format to enter for the target location. The primary example shows how to enter the target location if you are using a dedicated integration gateway. The

alternate example shows the format to use if the default local node points to a different gateway where WSDL documents and XSD schemas are available.

To define target locations:

1. On the Service Configuration page, click the **Set Target Locations** link.

   The Target Locations page appears.

2. In the Web Services Target Locations box, in the **Target Locations** field enter the target location, following the example format shown. Note that the port value you enter is the HTTP port.

3. In the REST Target Locations box, in the **Target Locations** field enter the target location of REST services, following the example format shown. Note that the port value you enter is the HTTP port.

4. Click the **OK** button.

   The Service Configuration page appears.

Defining Service System Status

The Services Configuration page contains a **Service System Status** drop-down list box that enables you to restrict rename, delete, and other administrative actions that users can perform on services, service operations, messages, and other integration metadata.

You can select one of two values from the drop-down list: *Production* or *Development.*

By default the status is set to *Development* and is the less-restrictive option in terms of enabling users to rename, delete and perform other actions on metadata. You may want to evaluate if the default setting is the appropriate one for your requirements and change it if necessary.

Located elsewhere in the product documentation is a table that describes the impact of the service system status on managing integration metadata.

Related Links

Configuring the Integration System to Handle Services

Activity 3: Registering Nodes with the Network

In this activity you:

- Set the network node password.
- Confirm that integration gateway keystore values are set.
- Register nodes in the network.

Understanding the Registering Nodes with the Network Activity

This section provides overview information about the tasks to perform in this activity.

Network Node Password

The first task in this activity is to set the network node password.

PeopleTools is delivered with an **IB_NETWORK** node. This node is used to perform functionality across all nodes in the integration network, including registering network nodes on participating systems. This node definition is delivered as an *Active* node definition. You can view the node definition in the Nodes component in PIA. Other than setting the password for this node, no additional configuration is required.

Note: Do not modify the IB_NETWORK node definition.

The network participants determine the password to be used and it must be set on each integration system.

Integration Gateway Keystore Values

The first activity of this guide, *Activity 1: Setting Up the Gateway,* includes the task of specifying the path to the gateway keystore and specifying the gateway keystore password in the integration gateway properties file.

This activity enables you to verify that the keystore path and password are set.

The importance of setting the keystore path and password in the integration gateway properties file cannot be stressed enough. If these values are not set, integrations will fail.

Registering Nodes with the Network

In this task you register remote PeopleSoft nodes that represent your integration partners in the integration network.

Note: Only PeopleSoft nodes can be registered in the integration network.

Registering nodes in the integration network enables you and your integration partners to take advantage of the features and options in the integration network, including view the status of nodes in the network, verify integration processing among network nodes, introspect and deploy integrations among network nodes, monitor integration and transaction processing, and more.

Prerequisites for Registering Nodes with the Network

For a node to be available to register in the network it must be:

- Defined in the database as an active *PIA* type node.
- Registered in the local gateway

Adding nodes to the database is a prerequisite for using this activity guide. Registering nodes in the gateway is a task in the first activity of this guide, *Activity 1: Setting up the Integration Gateway.*

Related Links

Prerequisites for Configuring Integration Broker Using the Activity Guide
Activity 1: Setting Up the Integration Gateway

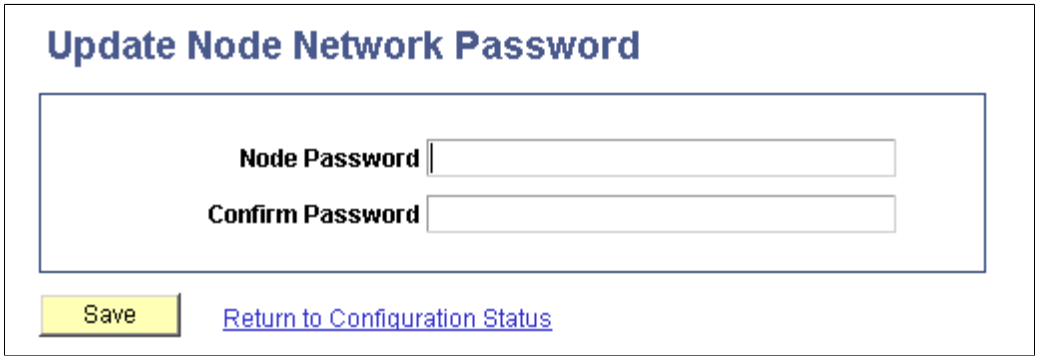
Setting the Network Node Password

Use the Update Node Network Password page to set the network node password.

Important! This password must then be set on each system participating in the network. The password set must be identical on all systems participating in the integration network.

At this time you have to navigate out of the activity guide to set this password. To access the page select **PeopleToolsIntegration BrokerIntegration Network WorkCenterConfiguration.**The Configuration Status page appears. In the Related Links section of the Configuration Status page, click the **Update Node Network Password** link.

This example illustrates the Update Node Network Password page.



To set the network node password:

1. Access the Update Node Network Password page.

   Select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **Configuration.**

   The Configuration Status page appears.

   On the Configuration Status page, locate the Related Links section at the bottom of the page and click the **Node Network Password** link.

2. In the **Node Password** field, enter a password.

3. In the **Confirm Password** field, enter the password again.

4. Click the **Save** button.

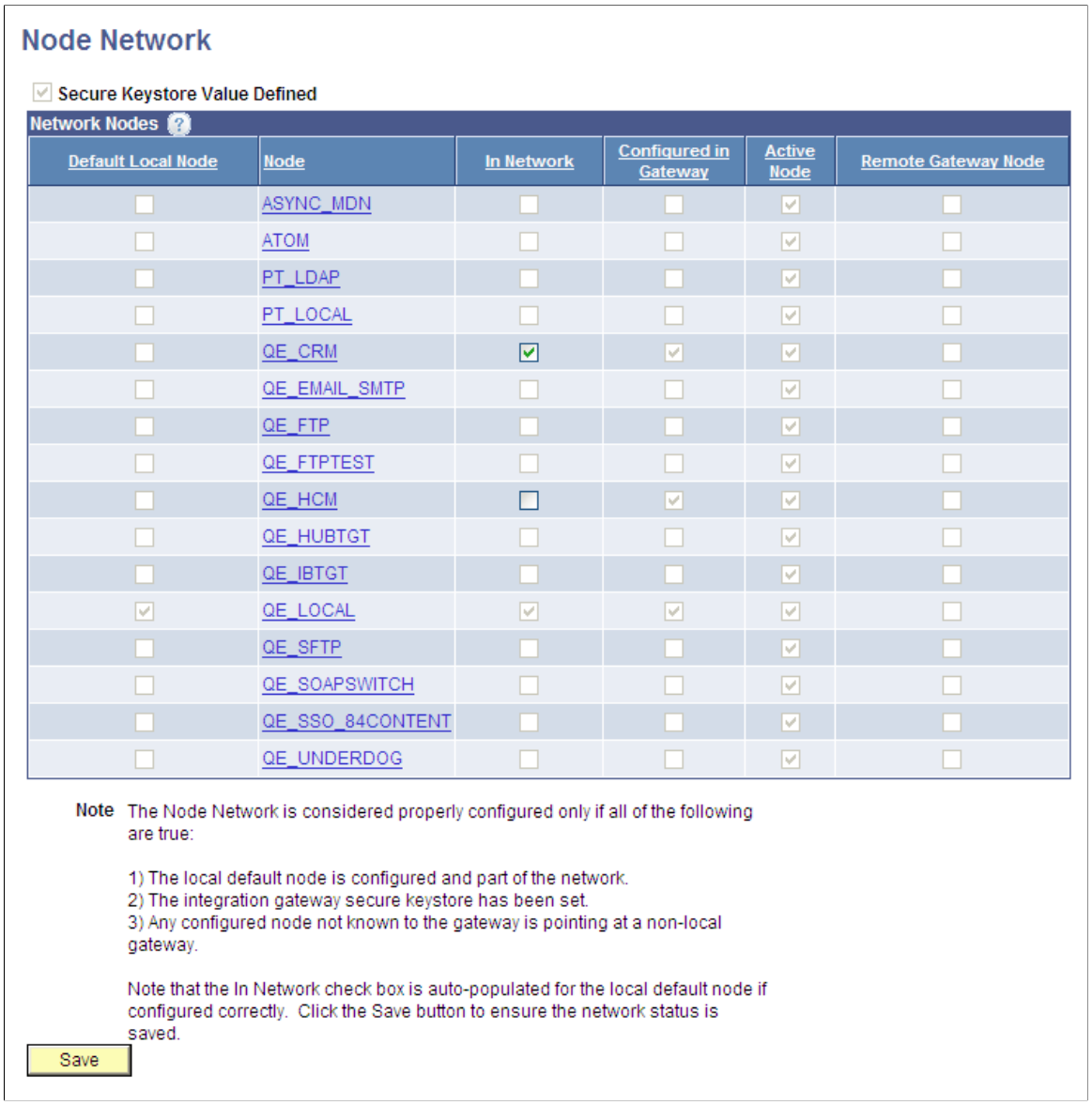
5. Navigate back to the IB Configure Network pagelet using the control in the navigation pane of the Integration Network WorkCenter.

Confirming Integration Gateway Keystore Values are Set

You can confirm that the path to the gateway keystore and the keystore password are set on the Node Network page. The page features a read-only **Secure Keystore Value Defined** field that indicates if these values are set in the integration gateway properties file.

To access the Node Network page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network** > **Register Node with Network.**

This example illustrates the Node Network page.



The top left corner of the Node Network page features a read-only **Secure Keystore Value Defined** field. When the box is selected, the path to the gateway keystore and the keystore password are defined in the integration gateway properties file. The previous example shows that the **Secure Keystore Value Defined** field is selected and indicates the values are defined in the integration gateway properties file.

Important! The importance of defining the keystore path and password in the integration gateway properties file cannot be stressed enough. If these values are not set, integrations will fail.

If on your system the **Secure Keystore Value Defined** field is not selected as shown in the example, return to first activity in this activity guide and define these values.

Related Links

Activity 1: Setting Up the Integration Gateway
Installing Digital Certificates on the Integration Gateway
Configuring Security and General Properties
Configuring the Integration Network

Registering Nodes with the Network

If you are integrating with nodes defined as PeopleSoft nodes register them in the integration network. PeopleSoft nodes are those nodes defined as *PIA* node types in their respective node definitions.

By default, the local default node is automatically registered in the integration network.

To register nodes in the network, use the Node Network page shown in the previous section.

The **In Network** control is enabled for all nodes that are available to register in the network. A check mark appears in the box to indicate the node is registered in the network.

In the previous example, the *QE_LOCAL* node and the *QE_CRM* node are registered in the network. The only other node eligible to be registered in the network is the *QE_HCM* node.

If you are integrating with PeopleSoft partners, they need to perform the task in this section and add to the integration network on their local system the node that they created on their local database to represent your system.

To register nodes in the network:

1. Access the Node Network page (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network** > **Register Node with Network**.)
2. Select the **In Network** box for each node to add to the
3. Click the **Save** button.

Related Links

Configuring the Integration Network

Activity 4: Activating Pub/Sub Server Domains

In this activity you will activate the publication/subscription (pub/sub) server domain. The pub/sub server processes in the pub/sub server domain enable the system to process asynchronous service operations.

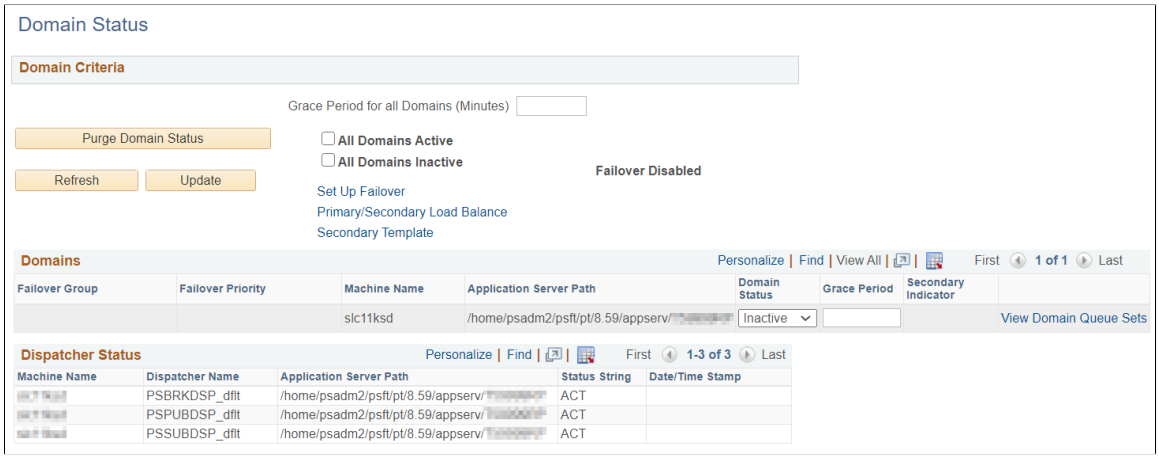
Prerequisites for Activating the Pub/Sub Server Domain

To activate the pub/sub server domains, the pub/sub server option in the PSADMIN domain configuration, **1.) Pub/Sub Servers,** must be set to *Yes*.

Activating the Pub/Sub Server Domain

Use the Domain Status page to activate the pub/sub server domain. To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network** > **Activate Domain.**

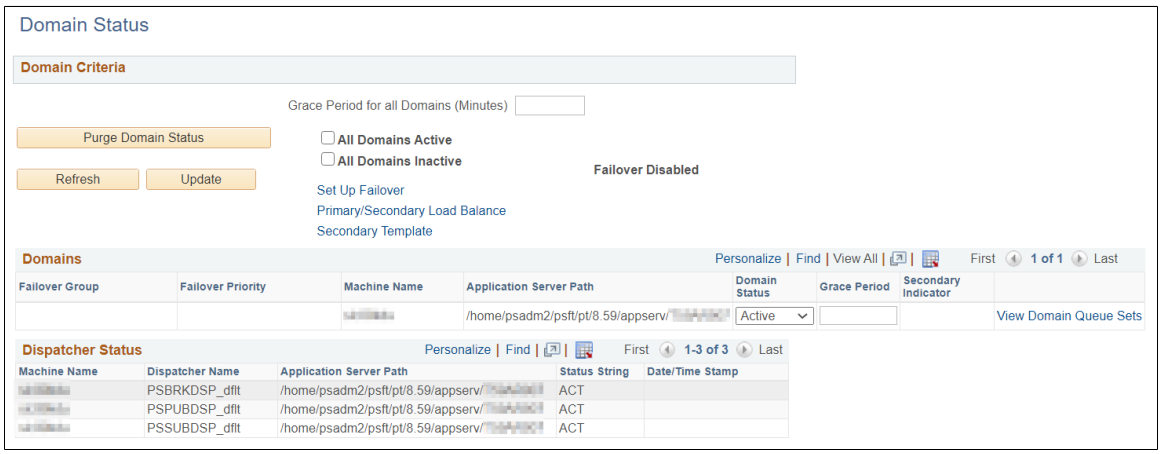
This example illustrates the Domain Status page before the pub/sub server domain is activated.



The Domains grid shows the machine name and the path to the application server on which the pub/sub server resides.

In the previous example, the pub/sub server domain is not active, as indicated by the *Inactive* value in the **Domain Status** field.

This example illustrates the Domain Status page when the pub/sub server domain is active.



When you activate the domain, as the previous example illustrates, the pub/sub dispatcher server processes appear in the Dispatcher Status grid with an active (*ACT*) status.

To activate the pub/sub server domain:

1. Access the Integration Broker Configuration activity guide (**PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network**.)

2. Click the Activate Domain link in the left navigation pane.

The Domain Status page appears.

3. From the **Domain Status** drop-down list, select *Active.*
4. Click the **Update** button.
5. Click the **Refresh** button.

Three dispatcher processes should appear in the Dispatcher Status grid with the status of active (*ACT*).

On occasion it may take a few moments for the processes to start. If the processes don't immediately appear in the grid, wait a few moments and click the **Refresh** button again.

If the three processes appear in Dispatcher Status grid, but with the status of inactive (INACT), click the **Update** button.

Related Links

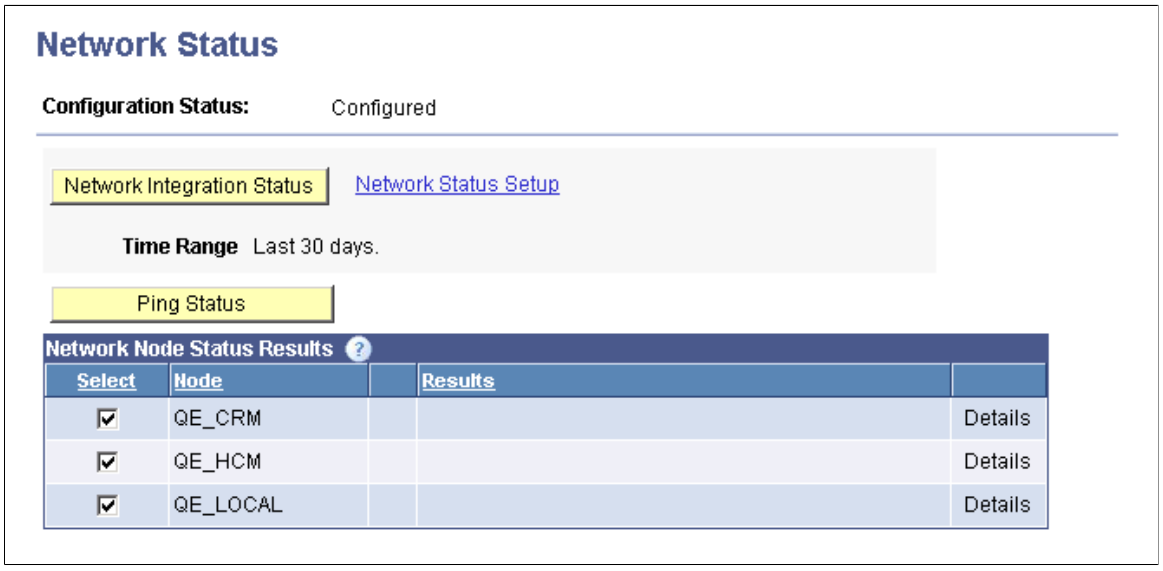
"Messaging Types" (Integration Broker)

Activity 5: Checking Network Connections

In this activity you will verify that the local node can communicate with other PeopleSoft nodes defined in the integration network.

Use the Network Status page to check network connections. To access the page select **PeopleTools** > **Integration Broker** > **Integration Network WorkCenter** > **IB Configure Network** > **Check Network Connections.**

This example illustrates the Network Status page.



The Network Node Status Results grid lists all of the nodes that are configured in the integration network. You can select one or all nodes to include in the connectivity check.

Click the Ping Status button to verify that the network nodes can communicate with one another.

A successful ping indicates that you have correctly defined the selected remote PeopleSoft nodes in the integration network and that the local system can connect to them. A successful ping also indicates that the PeopleSoft integration partners represented by the remote nodes have correctly defined your node as part of their integration network and they should be able to perform a successful integration network ping to your system as well.

See Pinging Integration Network Nodes for complete information on using this page.

Activity 6: Introspecting and Deploying Integrations

In this activity you introspect and deploy integrations among the integration partners defined in the network.

Note: If you are using this activity guide to perform a basic configuration of PeopleSoft Integration Broker as a prerequisite to use another PeopleTools technology you may not need to perform this activity. Consult the product documentation for the PeopleTools technology that you are using to determine if this activity is required.
If you and your integration partners do not have any integrations developed or if you are integrating with third-party systems, you do not need to perform this activity.

If you determine that you do not need to introspect and deploy integrations, mark this activity complete in the activity guide.

Prerequisites for Performing the Introspection and Deployment Activity

To perform the introspection and deployment activity:

- You should have a basic understanding of the integration metadata and definitions required for integrations. These metadata and definitions include services, service operations, handlers and routings.
- Service operations for integrating with your integration partners must exist.

Related Links

“PeopleSoft Integration Broker Metadata” (Integration Broker)

Performing the Introspection and Deployment Activity

The Integration Network features a series of pages that enable you to introspect and deploy services on remote PeopleSoft nodes. The system can introspect and deploy integrations on any node defined in the integration network.

The introspection process makes a series of checks on service metadata to ensure that metadata definitions exist, that they are complete, and that they are active. For example, the introspection process checks that for a selected service or service operation that the service operation is active, routing definitions exist, handlers are active, service operation permissions are defined, and so on.

The goal of deployment processing is to resolve any issues identified during introspection. In many cases during deployment processing, the system can activate service operations, handlers, and routings

that are found to be inactive during the introspection process. The system can also frequently create routings if none are found to exist during introspection. However, there are some situations where manual intervention is required. Manual intervention may be required to resolve routing issues and handler issues. And in all cases where service operation permissions do not exist, manual intervention is required to assign permissions.

The basic steps for introspecting and deploying integrations are:

- Search for items to introspect and deploy.
- Select integrations to introspect and deploy.
- Select nodes for introspection and deployment.
- Verify nodes and services to introspect.
- Viewing results.

Introspecting and deploying integrations is documented elsewhere in the product documentation.

Related Links

Introspecting and Deploying Network Integrations

Activity 7: Updating Security on Service Operations

In this activity you set security permissions on service operations.

Note: If there are no service operations developed and ready for testing or production environments, you do not need to perform this activity.
This activity pertains to setting service operation permissions for integrations among PeopleSoft nodes only.

If you determine that you do not need to perform this activity, mark it complete in the activity guide.

Prerequisites for Updating Security on Service Operations

To update security on service operations, service operations must exist on the local database.

Updating Security on Service Operations

To assign and update permissions on service operations, you assign one or more permission lists and access levels to the service operations.

The Integration Network features pages that enable you to assign and update permissions on individual service operations or in bulk to service operations.

Setting service operation permissions is describe in detail elsewhere in the product documentation.

Related Links

Setting Service Operation Permissions

Chapter 14

Setting Up Secure Integration Environments

Understanding Setting Up Secure Integration Environments

This topic provides an overview of securing integration environments, outbound PeopleSoft Integration Broker security processing, and inbound PeopleSoft Integration Broker security processing. This topic also features code examples to help illustrate security concepts and Integration Broker security features.

Note: The code examples in this topic are for illustrative purposes only and are not intended to be used in a production environment.

Understanding Securing Integration Environments

This section discusses types of integration security and provides an overview of security terminology used in conjunction with PeopleSoft Integration Broker.

Web Server SSL/TLS Encryption

Encryption supports data privacy. When encryption is implemented, the sender translates the content of a transaction into a secret code that only the receiver can decrypt. PeopleSoft Integration Broker supports the Secure Sockets Layer (SSL) protocol and Transport Layer Security (TLS) protocol for data encryption.

Note: The TLS security protocol is the successor to the SSL security protocol. The steps for setting up SSL and TLS are similar, and hence are referenced as “SSL/TLS” in this topic. However, it's important to note that while these protocols are similar, they do not interoperate.

You can employ SSL/TLS encryption at the web server level to secure data sent between your web server and that of your integration partners.

You can implement web server SSL/TLS encryption with integration partners running on all PeopleTools 8.4x systems and third-party systems.

You use digital certificates to implement SSL/TLS encryption.

WS-Security

Web services security (WS-Security) is implemented on the integration gateway for inbound and outbound integrations with third-party systems.

You can implement WS-Security using username tokens or Security Assertion Markup Language (SAML) tokens .

You can implement WS-Security with integration partners running on PeopleTools 8.48 and later systems and third-party systems.

WS-Security using Username Token Profile

The WS-Security Username Token Profile defines a standard way of identifying the requestor by “username”, and optionally using a password (or shared secret, or password equivalent) to authenticate that identity to the web service producer.

On outbound request processing, PeopleSoft Integration Broker generates a WS-Security UsernameToken, which may include a password. The WS-Security information is added to the SOAP request on the integration gateway prior to sending to the integration partner.

On inbound processing, PeopleSoft Integration Broker can process requests received from integration partners that contain WS-Security UsernameToken and password in the SOAP header of the inbound SOAP request.

WS-Security using SAML Token Profile

The SAML Token Profile uses assertions to define a standard way to associate common information such as issuer ID, assertion ID, subject and so on.

On outbound request processing, PeopleSoft Integration Broker adds a WS-Security SOAP header to the service operation that contains SAML credentials defined in the node definition for the node.

On inbound processing, the PeopleSoft system checks for the existence of a WS-Security SOAP header. If it exists, the integration gateway decrypts the SAML token (if it has been encrypted) to restore the user ID information to clear text format.

Related Links

“Understanding WS-Security” (Security Administration)

Client Authentication

Outbound requests connect from the application server to the integration gateway using an MIME over HTTP connection. To secure the connection you can employ client authentication. This option is typically implemented when the application server and integration gateway reside on separate machines. Client authentication is used only on outbound transactions, since inbound transactions connect between the integration gateway and application server are made using Jolt connection strings.

Note: If you implement client authentication you must also implement web server SSL encryption.

You can implement client authentication with integration partners running on all PeopleSoft 8.4x systems and third-party systems.

Nonrepudiation

Nonrepudiation is a form a digital security that ensures that a transferred message has been sent and received by the parties claiming to have sent and received the message. It is also a method of guaranteeing that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

You can implement nonrepudiation with integration partners running on all PeopleSoft 8.4x systems and third-party systems.

User Authentication

Service operations are secured at the user level. On an outbound transaction, user authentication sets the user ID to assign to the service operation.

When user authentication is implemented a user ID or user ID and password are required.

For inbound transactions, user authentication determines the user ID associated with the inbound service operation. If a user ID and password are required to invoke a service operation, the system validates the user ID to see if it is a member of the permission list to which the service operation is assigned.

You can implement user authentication with integration partners running on PeopleSoft 8.48 and later systems and third-party systems.

Node Authentication

Use node-level security for integrations with nodes running on earlier PeopleTools 8.4x releases.

To implement node-level security you define an authentication option for the node using the Nodes page. You can use a node certificate or a password as authentication options.

Node-level security pertains to inbound and outbound processing and authentication is performed on the application server.

You can implement node authentication with integration partners running on all PeopleSoft 8.4x systems and third-party systems.

Service Operation Permission Lists

The user ID that is authenticated during user authentication is validated against the permission list to which the service operation is assigned.

Understanding PeopleSoft Integration Broker Security Processing

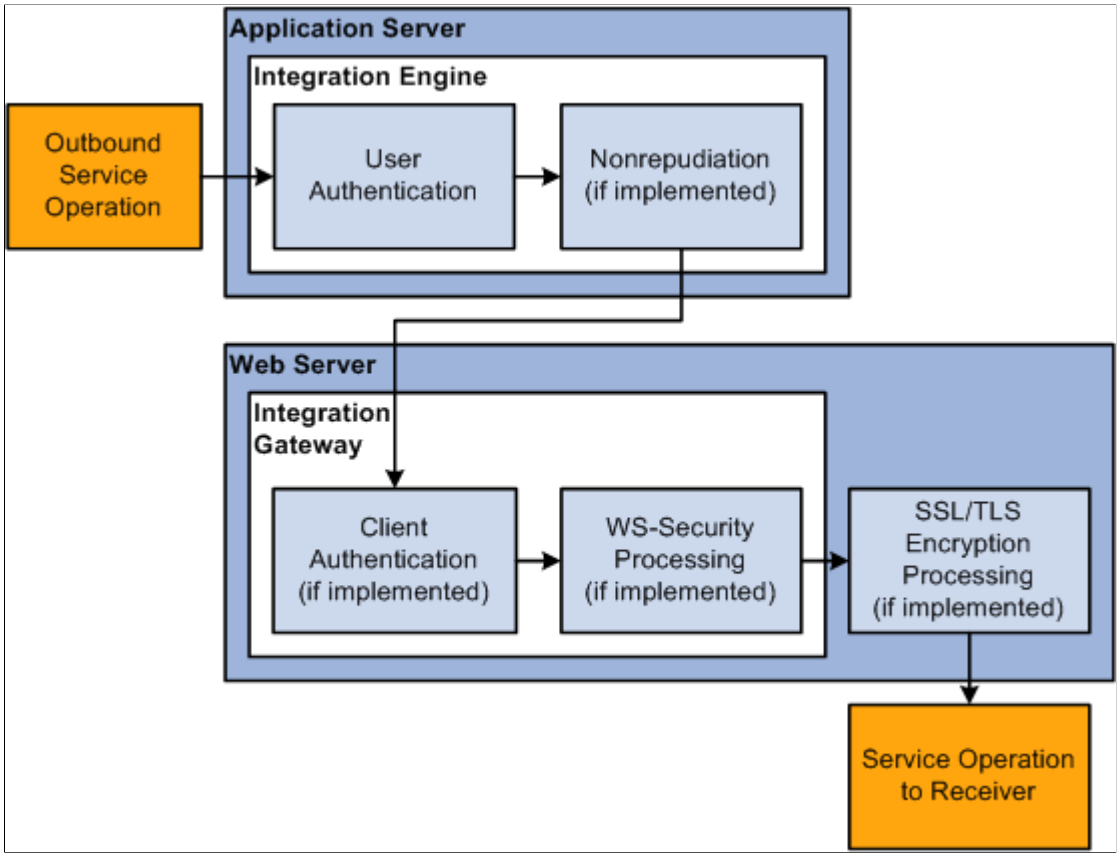
This section discusses:

- Outbound PeopleSoft Integration Broker security processing.
- Inbound PeopleSoft Integration Broker security processing.

Outbound Integration Broker Security Processing

This section discusses security processing for outbound integrations from PeopleSoft Integration Broker.

This diagram illustrates security processing for outbound integrations from PeopleSoft Integration Broker. The diagram shows that as an outbound service operation passes onto the application server, it goes through user authentication and nonrepudiation (if implemented) on the integration engine. As processing moves to the web server, the service operation may go through client authentication (if implemented) and WS-security processing (if implemented). The last type of security processing through which a service operation may go is through on the PeopleSoft side is SSL/TLS encryption (if implemented) on the web server.



PeopleSoft Integration Broker applies the following security elements to outbound integrations:

Note: The elements are discussed in the order in which the system applies them.

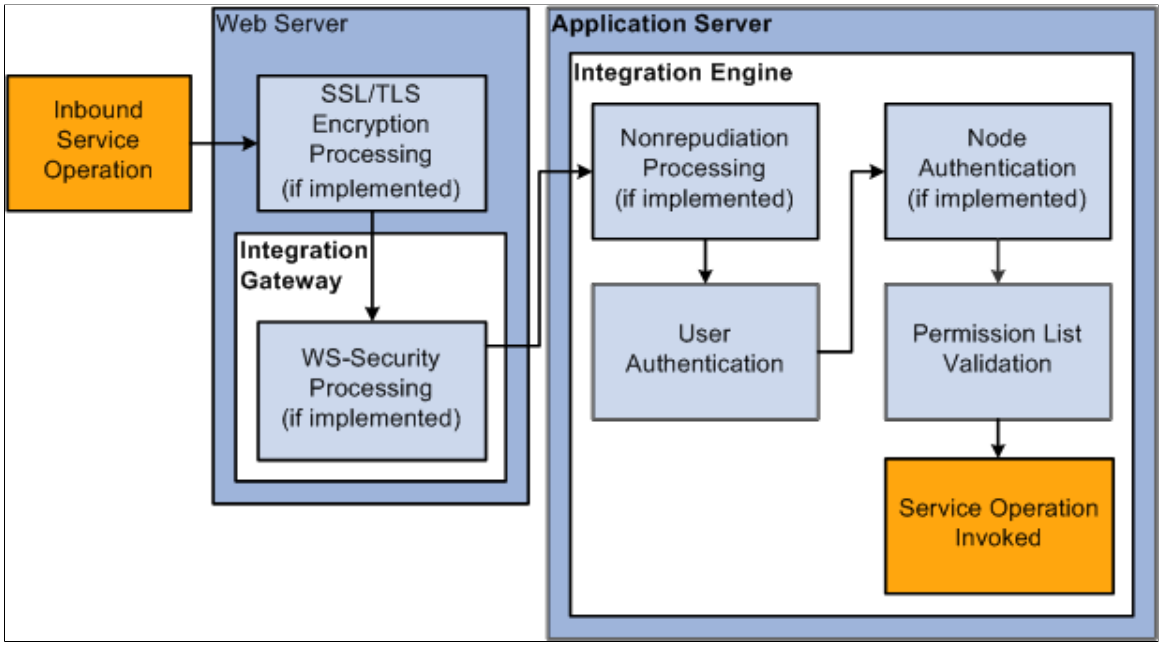
Term	***Definition***
User authentication	If the outbound service operation originates from a PeopleSoft (*PIA*) node, the user authentication process attaches the PeopleSoft authentication token to the service operation. If the service operation originates from an external (External) node, the model determines the user ID for the service operation and passes the information to the WS-Security framework so it can generate the UsernameToken for the outbound transaction.
Nonrepudiation	Nonrepudiation processing is performed.

Term	*Definition*
Client authentication	Client authentication secures the connection between the PeopleSoft application server and the integration gateway on outbound transactions. You use digital certificates to secure this connection.
WS-Security	Outbound WS-Security processing includes generating the UsernameToken for the WS-Security SOAP header. This process may also involve encrypting and digitally signing the data, if specified in the WS-Security parameters on the node.
SSL/TLS encryption	SSL/TLS encryption on outbound integrations establishes a secure web server connection with an integration partner.

Inbound Integration Broker Security Processing

This section discusses inbound integration broker security processing.

This diagram illustrates security processing for inbound integrations to PeopleSoft Integration Broker. The first type of security processing that may be performed on an inbound service operation occurs on the web server where SSL/TLS encryption processing takes place if it is implemented. As service operation processing moves to the integration gateway located on the web server, it may go through WS-security processing (if implemented). As processing moves to the application server, an inbound service operation my go through nonrepudiation processing (if implemented), then user authentication. After user authentication, node authentication (if implemented) occurs. The final security processing before service invocation is permission list validation.



PeopleSoft Integration Broker applies the following security elements to inbound integrations:

Note: The elements are discussed in the order in which the system applies them.

Term	*Definition*
SSL/TLS encryption	If the inbound service operation is encrypted, the integration gateway decrypts the data.
WS-Security	On inbound transactions, WS-Security processing includes validating a digital signature (if required), decrypting user information (if required), and passing the extracted user information to the integration engine for authentication.
Nonrepudiation	Nonrepudiation processing is performed.
User authentication	The system determines and validates the user ID associated with the inbound service operation.
Node authentication	If a node password is employed, the system validates that the inbound service operation contains the node password. If certificate authentication is employed, the system authenticates the node certificate.
Permission list validation	The system matches the user ID passed in with the service operation to the appropriate permission list.

Understanding Digital Certificates

This section provides an overview of :

- Digital certificates.
- Digital certificate authorities.
- Digital certificate installation elements.

Digital Certificates

A *digital certificate* is a form of electronic ID card that supports public key encryption technology. Each messaging participant generates a matched pair of encryption keys—a *private key,* which is never revealed or transmitted, and a *public key,* which is freely available to other participants. These keys are stored in a local file or repository called a *keystore,* and the public key is stored as part of a digital certificate. The certificate can be attached to a service operation to verify the sender's identity and to provide the recipient with the means to encode a response.

The following table lists the security technologies that require digital certificates and the digital certificate installation location for each of them. The table also lists the section in this topic that discusses installing digital certificates for each of the technologies:

Security Technology	*Digital Certificate Installation Location*	*Section Describing How to Install Digital Certificates*	*Comments*
SSL/TLS encryption.	Web server.	Setting Up Web Server SSL/TLS Encryption.	Secures web server-to-web server connections.
WS-Security.	Integration gateway.	Installing Integration Gateway-Based Digital Certificates.	Secures web server-to-web server connections.
Client authentication.	Integration gateway.	Installing Integration Gateway-Based Digital Certificates.	Secures application server-to-integration connections.
Nonrepudiation.	Application server.	Installing Application Server-Based Digital Certificates.	Authenticates sender and receiver.
Certificated-based node authentication.	Application server.	Installing Application Server-Based Digital Certificates.	Authenticates sender.

Digital Certificate Authorities

A certificate authority (CA) is a trusted third-party organization or company that issues digital certificates used to create digital signatures and encryption keys. The role of the CA in this process is to guarantee the identity of the party granted the certificate. Usually, this means that the CA has an arrangement with a financial institution that provides information to validate the grantee's identity.

To install digital certificates for secure messaging, you must select a CA from whom to obtain the certificates. There are many CAs to choose from, and most of them do business on the World Wide Web. Some of the best known are:

- Verisign, Inc.
- Entrust Technologies.
- Baltimore Technologies.
- Thawte.

There are also numerous lesser known CAs, which might be appropriate if they are well known in a particular geographical region or industry. One of the systems participating in a secure integration might even serve as CA for the other participants. Each CA provides a unique set of security services and has its own way of handling digital certificates.

Before you implement secure messaging with PeopleSoft Integration Broker, investigate the available CAs, select one or more from whom you will obtain digital certificates, and familiarize yourself with their policies and procedures.

Digital Certificate Installation Elements

Whether you implement digital signature authentication, nonrepudiation, or SSL encryption, you need to use digital certificates. Although these security features require you to use a variety of programs and procedures, some characteristics of digital certificates—including the process of obtaining, installing, and configuring them—are common to all three features.

Depending on the security feature, you might install digital certificates in the keystore of an application server, a web server, or an integration gateway. An implementation of digital certificates on each of these entities involves the following elements:

- The entity's private and public encryption keys.
- A distinguished name (DN) for the entity.
- A certificate signing request (CSR).
- A certificate containing the entity's public encryption key, signed by a trusted CA.
- A root certificate from the trusted CA.

The following sections discuss these elements in more detail.

Public and Private Encryption Keys

For a given keystore, you generate private and public encryption keys simultaneously as a matching pair with software provided by the entity.

DN for the Entity

A DN is a property commonly used in security environments to uniquely identify a person, system, or network node. The DN is usually stored as a string of name-value attribute pairs separated by commas and spaces. You must provide the DN attribute values to generate a private key. These attributes include:

Term	***Definition***
Common name (CN)	The name of the entity, expressed as a machine name, domain name, node name, or a name that you create, depending on the environment; for example, *QE_LOCAL*.
Organization unit (OU)	The part of the organization to which the entity belongs; for example, *Accounts Receivable*.
Organization (O)	The name of the organization or company; for example, *PeopleSoft*.
Locality (L)	The city or equivalent locality of the organization; for example, *Pleasanton*.

Term	*Definition*
State (ST)	The state, province, or equivalent region of the locality; for example, *California*.
Country (C)	The country of the locality; for example, *US*.

CSR

A certificate signing request, or CSR, is a document that contains the entity's public key. The CSR is typically generated in Privacy Enhanced Mail (PEM) format, which is base64–encoded binary data. PEM is a standard text-based format for storing and transmitting digital certificates. You use the same software to generate the CSR that you use to generate the private-public key pair. The following example shows a CSR:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBkTCB+wIBADBSMQswCQYDVQQGEwJ1czELMAkGA1UECBMCY2ExDTALBgNVBAcTBGhlcmUxCzAJ
BgNVBAoTAndlMQ0wCwYDVQQLEwR1bml0MQswCQYDVQQDEwJtZTCBnzANBgkqhkiG9w0BAQEFAAOB
jQAwgYkCgYEApaGAHNBjuByh8qXFCz33TgLzUjRm8S6tijit7fw23rKWyipQ0VgqeAD6eHr0pini
lyJPPOiJJ5fY0h2h78hOr8o+nJosTcqZL3jP+rSVick7qPPyXjcxP1UCGz/8RNykFDnbwjziwi+p
MesoWa8hfBss0ga2zZsmlV8Q4SyYE3UCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBACt0owTCngrU
/HAMAZgT/2O6hiZaD4OVBrgLYzmRvUiVhKOyTUzUv57ks7U6DQYt+rnWwNJtVbeAqO5eZiT7hXbj
Pwl8lGj+Adb6FGYOt4OhicZ0gNMHtURVop6iNJ9scxOmVcpkO0yX5f1rWFdZ0KZrWZSFGI6Lwdud
Hvbyvbpz
-----END NEW CERTIFICATE REQUEST-----
```

Signed Public Encryption Key From CA

The process of obtaining a signed public key certificate from a CA depends on the CA that you select. Typically, it requires you to paste the content of the PEM-formatted CSR into a form that you submit online. The CA then creates, digitally signs and returns a public key certificate to you. The CA will either email you the certificate or require you to download it from a specified web page. The certificate can be either PEM or the binary Distinguished Encoding Rules (DER) format. Following is an example of a PEM-formatted certificate:

```
-----BEGIN CERTIFICATE-----
MIICIDCCAcqgAwIBAgIQrDVQJKAAKLQR0/bIDJMSVDANBgkqhkiG9w0BAQQFADBy
MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExEzARBgNVBAcTClBsZWFzYW50b24x
FzAVBgNVBAoTDlBlb3BsZVNvZnQgSW5jMRMwEQYDVQQLEwpQZW9wbGVUb29sMRMw
EQYDVQQDEwpQZW9wbGVUb29sMB4XDTAwMDMxMDIxMTIzNVoXDTA1MDMxMDIxMTIz
NVowcjELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNBMRMwEQYDVQQHEwpQbGVhc2Fu
dG9uMRcwFQYDVQQKEw5QZW9wbGVTb2Z0IEluYzETMBEGA1UECxMKUGVvcGxlVG9v
bDETMBEGA1UEAxMKUGVvcGxlVG9vbDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCy
o44wplb57M272GRP3sC4TtLm/MD1G9osRjG9BWnsjjTij9GNi6Rnf9cNxkj+AGQY
gnE3P7lp9rYN6GQxPldNAgMBAAGjPDA6MAsGA1UdDwQEAwIBxDAMBgNVHRMEBTAD
AQH/MB0GA1UdDgQWBBSkFZJ1Dtt5uE6muLRN3rwRPsUCsTANBgkqhkiG9w0BAQQF
AANBAJec3hFPS2SLSDtfLI9mSA7UL1Vgbxr5zZ4Sj9y4I2rncrTWcBqj7EBp9n/Z
U/EwDEljVbE8SSDYr1Emgoxsr4Y=
-----END CERTIFICATE-----
```

Root Certificate

The root certificate contains the CA's digitally signed public key. It's also known as a *chain file* or a *signer certificate*. The process of obtaining a root certificate from a CA depends on the CA. The CA typically sends an email with the certificate or requires you to download it from a specified page.

Note: PeopleSoft systems accept root CA's with key sizes up to 4096 bits.

The signed public key certificate also contains an embedded copy of the CA's root certificate, which you can export.

Installing Integration Gateway-Based Digital Certificates

This topic provides an overview of integration gateway-based digital certificates and discusses how to:

- Generate private and public key pairs.
- Generate CSRs.
- Obtain signed root certificates.
- Import signed root certificates.
- Specify the keystore location for WS-Security in the wss.properties file..
- Encrypt keystore passwords for WS-Security.

Understanding Integration Gateway-Based Digital Certificates

Use the procedures discussed in this section for generating and installing digital certificates for use with the following security protocols:

- Integration gateway encryption.
- Client authentication.
- WS-Security.

Elements of Integration Gateway-Based Digital Certificates

To set up integration gateway-based digital certificates, use the PSKeyManager utility to install digital certificates in the integration gateway keystore.

The integration gateway requires the following elements:

- The gateway's private key.
- A certificate containing the gateway's public key, digitally signed by a trusted CA.
- A root certificate from the CA that signed the gateway's public key.

Using the PSKeyManager utility, you generate a private-public key pair, which is automatically inserted in the gateway keystore.

You generate a PEM-formatted CSR that contains the gateway's public key. You submit the CSR to the selected CA. The CA creates, digitally signs, and returns your gateway's public key certificate to you. This certificate also contains a signed copy of the CA's root certificate. These certificates may be in standard DER-encoded binary format, or they can be converted to PEM format if necessary.

You then install both signed certificates in the gateway keystore. In addition, you register them and the private key with the web server so that it can recognize and use them.

Keystore Location for Integration Gateway-Based Digital Certificates

The keystore location for integration gateway-based certificates is:

```
<PIA_HOME>\webserv\peoplesoft\piaconfig\keystore
```

Integration gateway, client authentication, and WS-Security certificates are stored in this location.

In addition, SSL/TLS digital certificates are also stored in this location.

wss.properties File

The wss.properties file stores keystore location information and password information for WS-Security digital certificates.

When installing digital certificates for WS-Security, you must specify the location of the keystore in this file.

You can also store an encrypted copy of the keystore password in this file.

The location of the file is:

```
<PIA_HOME>\webserv\<DOMAIN>\peoplesoft\applications\PSIGW.war\WEB-INF\classes
```

Generating and Installing Integration Gateway-Based Certificates

Use the information provided in the topic “Installing Web Server-Based Digital Certificates” to generate and install integration gateway-based certificates.

Note that for integration gateway encryption if the integration gateway is installed on a web server that has SSL/TLS implemented, the integration gateway and web server can share the same digital certificates. As a result, you do not need to install separate integration gateway certificates. However, if the integration gateway is installed on a web server where SSL/TLS is not implemented, you must generate and install digital certificates on that web server.

After generating and installing integration gateway-based certificates the keystore path and the keystore password must be defined in the integration gateway properties file.

Warning! Integrations will fail if the keystore path and encrypted keystore password are not defined in the integration gateway properties file, integrationGateway.properties.

If you are implementing WS-Security you must specify the keystore location in the wss.properties file. The next section describes how to specify the keystore location in the wss.properties file.

Related Links

Configuring Security and General Properties

Specifying the Keystore Location for WS-Security in the wss.properties File

After you install digital certificates for WS-Security, you must specify the keystore location in the wss.properties file.

To specify the keystore location for WS-Security:

- Open the wss.properties file.

  The location of the file is <PIA_HOME>\webserv\<DOMAIN>\peoplesoft\applications\PSIGW.war\WEB-INF\classes.

- Set the following property equal to the location and file name of the keystore where you installed the integration gateway-based digital certificates.

  ```
  org.apache.ws.security.crypto.merlin.file
  ```

  For example:

  ```
  org.apache.ws.security.crypto.merlin.file=c:/<PIA_HOME>/<webserv>/
  <DOMAIN>/keystore/pskey
  ```

  Note: When entering the path to the keystore, use either double backslashes (“\\”) or forward slashes (“/”) as path separators. Do not use backslashes (“\”) as path separators for directory names in the wss.properties file. Backslashes are misinterpreted as escape characters by the Java processes that access the file.

- Save the changes.

Encrypting Keystore Passwords for WS-Security

This section discusses how to encrypt the password for the keystore that contains digital certificates for WS-Security.

Understanding Encrypting Keystore Passwords for WS-Security

When working with the WS-Security digital certificates, PeopleSoft recommends that you encrypt the keystore password in the wss.properties file using the PSCipher utility.

Encrypting the WS-Security Keystore Password

To encrypt the WS-Security keystore password, making sure to write down the encrypted output.

1. Encrypt the WS-Security keystore password using the PSCipher utility.

   See Encrypting Passwords Using the PSCipher Java Utility.

2. Access the wss.properties file.

   The location is <PIA_HOME>\webserv\<DOMAIN>\peoplesoft\applications\PSIGW.war\WEB-INF\classes.

3. Set the following property equal to the encrypted password you created using the PSCipher utility:

   ```
   org.apache.ws.security.crypto.merlin.keystore.password
   ```


The following example shows an encrypted password entered for this property:

```
org.apache.ws.security.crypto.merlin.keystore.password== *** Encrypted passwor⇒
d ***
```

4. Save the changes.

Implementing Web Services Security

This section provides and overview of WS-Security and WS-Security processing in PeopleSoft Integration Broker. It also discusses prerequisites for implementing WS-Security in PeopleSoft Integration Broker and discusses how to:

- Implement WS-Security for inbound integration (Username Tokens).
- Implement WS-Security for inbound integration (SAML Tokens).
- Implement WS-Security for outbound integration (Username and SAML Tokens).
- Override node-level WS-Security settings on routing definitions.
- Implement WS-Security on services consumed using the Consume Web Services wizard.

This section also describes WS-Security configuration options for outbound integrations and provides examples for WS-Security SOAP message headers.

Understanding Implementing WS-Security in PeopleSoft Integration Broker

This section provides an overview of implementing WS-Security in PeopleSoft Integration Broker.

WS-Security Standard Supported

PeopleSoft implements WS-Security in accordance with Oasis standards.

Within this framework, PeopleSoft implements:

- Username tokens.
- SAML tokens.

The PeopleSoft implementation of WS-Security supports:

- Clear-text username token. (Password is optional.)
- Digitally signed username token.

  Digital signatures apply to the SOAP message header and SOAP message body.
- Encrypted username token.

  You specify to encrypt the SOAP header only, SOAP header and message body, or the message body only

- Encrypted SAML assertion token.

  You specify to encrypt the SOAP header only, SOAP header and message body, or the message body only

Please visit the My Oracle Support website for information about the current versions of the WS-Security standards, profiles, and namespaces supported by PeopleTools.

UsernameToken Profile

A UsernameToken is the means of identifying a requestor by user name to authenticate the user's identity to the web service provider. A password may also be used in conjunction with the user name.

The UsernameToken is supplied in the <UsernameToken> element in the WS-Security SOAP header that gets added to an inbound or outbound service operation when WS-Security is implemented. The elements included in the credential are discussed in the following section.

On outbound service operations, the values that the PeopleSoft system populates in the UsernameToken profile can be derived from an external user ID that you specify on the node definition for the external node. It can also be derived from the default user ID specified on the external node definition. In addition, you can choose to digitally sign and encrypt this information.

SAML Token Profile

The Security Assertion Markup Language (SAML) is an XML-based framework for exchanging security information. All SAML tokens include the following common information as defined by Oasis standards:

- Issuer ID.
- Subject.
- Name.
- Subject confirmation.
- Conditions under which the assertion is valid.

  Example of these conditions are *NotBefore* and *NotOnOrAfter*.

This security information is expressed in the form of assertions about subjects, where a subject is an entity that has an identity in some security domain.

The following pseudocode shows an example of a SAML token:

```
<Assertion AssertionID="d9aeaa4c1126df5ee0c6df64fdf961b1" IssueInstant="2008-05-
  14T18:18:47.246Z" Issuer=".example.com" MajorVersion="1" MinorVersion="1"
  xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml="urn:oasis:names:tc:
  SAML:1.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
  <Conditions NotBefore="2008-05-14T18:18:47.184Z" NotOnOrAfter="2008-05-14T18:
   28:47.184Z"/>
   <AuthenticationStatement AuthenticationInstant="2008-05-14T18:18:47.215Z"    Aut⇒

henticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
      <Subject>
        <NameIdentifier NameQualifier=".example.com">QEDMO</NameIdentifier>
         <SubjectConfirmation>
           <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
           </ConfirmationMethod>
         </SubjectConfirmation>
```



```
      </Subject>
   </AuthenticationStatement>
</Assertion>
```

Note these points about PeopleSoft SAML assertions:

- The PeopleSoft SAML token is concerned with the authentication statement only.
- The PeopleSoft SAML token supports SAML with digital signature and encryption. SAML tokens without digital signatures are not supported.
- The PeopleSoft SAML profile of WSS: SOAP Message Security requires that systems support sender-voucher methods of subject confirmation.
- The SAML Assertion validity or condition by default is set to 10 minutes. However, you can override the default time by adding org.apache.ws.security.saml.AssertValidMins=15 in the wssSAML.properties file which is located in the \\WEB-INF\classes\wssSAML.properties directory.

WS-Security SOAP Header

Inbound and outbound transactions that are secured with WS-Security pass the security credentials in a WS-Security SOAP header that is added to the service operation.

The following elements can appear in the WS-Security SOAP header generated by the integration gateway:

Element	***Description***
<wsse:UsernameToken>	Username and optional password to authenticate.
<wsse:Username>	Username to use for authentication. On outbound integrations this name can be generated using the External User ID or Default User ID values that you define on the node definition. In addition, you can select to digitally sign and encrypt this value.
<wsse:Password>	(Optional.) Password for the authentication username. On outbound integrations this password matches that specified for the External User Id or Default User ID used to generate the username. If you select to digitally sign or encrypt the username, this password is digitally signed or encrypted as well.
<saml:Assertion>	SAML assertion token to use for authentication. You can encrypt this value.

The following example shows a WS-Security SOAP header for an outbound service operation generated by the PeopleSoft system:

```
<soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse=
          "http://docs.oasis-open.org/wss/2004/01/oasis-200401-
            wss-wssecurity-secext-1.0.xsd">
            <wsse:UsernameToken>
                <wsse:Username>PTDMO</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/
                  wss/2004/01/oasis-200401-wss-username-token-profile-
                   1.0#PasswordText">PTDMO</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
```

SAML assertions and references to assertion identifiers are contained in the <wsse:Security> element, which in turn is included in the <SOAP-ENV:Header> element. The following example shows SAML assertions conveyed within a WS-Security header as part of a SOAP message:

```
<SOAP-ENV:Envelope>
   <SOAP-ENV:Header>
     <wsse:Security>
       <saml:Assertion> - - - </saml:Assertion>
     </wsse:Security>
   </SOAP-ENV:Header>
  <SOAP-ENV:Body> - - - </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Understanding WS-Security Processing using Username Tokens

This section provides overviews of:

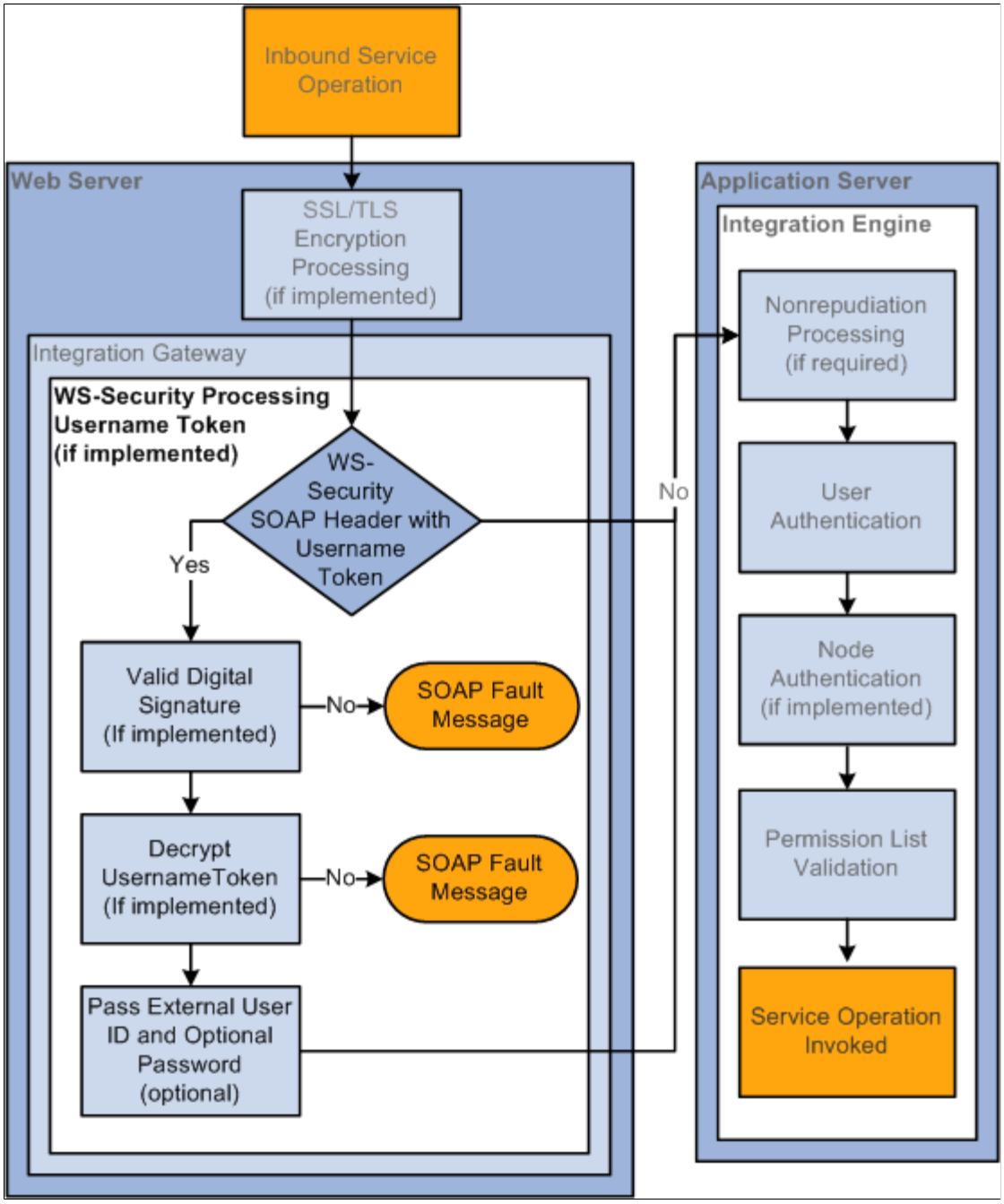
- Inbound WS-Security processing using Username tokens.
- Outbound WS-Security processing using Username tokens.

Inbound WS-Security Processing using Username Tokens

The inbound processing of service operations that are WS-Security-compliant using Username tokens occurs on the integration gateway.

This diagram illustrates inbound WS-Security processing in PeopleSoft Integration Broker when it is implemented. The diagram shows all possible security processing for an inbound integration to show where in the processing flow WS-Security processing occurs. WS-Security processing is highlighted in the foreground of the diagram.

The diagram illustrates the steps in WS-Security processing, including validation that the service operation contains a WS-Security SOAP header with a username token, validation of a digital signature if implemented, decryption of the username token if encrypted, and passing the external user ID and password to the application server.



When any transaction arrives at the integration gateway, the PeopleSoft system checks for the existence of a WS-Security SOAP header. If it exists, the integration gateway validates the digital signature if it exists,

and decrypts the UsernameToken and optional password to restore the user ID information to clear text format.

The integration gateway then passes the user ID information, and UsernameToken password if provided by the sender, to the application server, where additional security processing is performed.

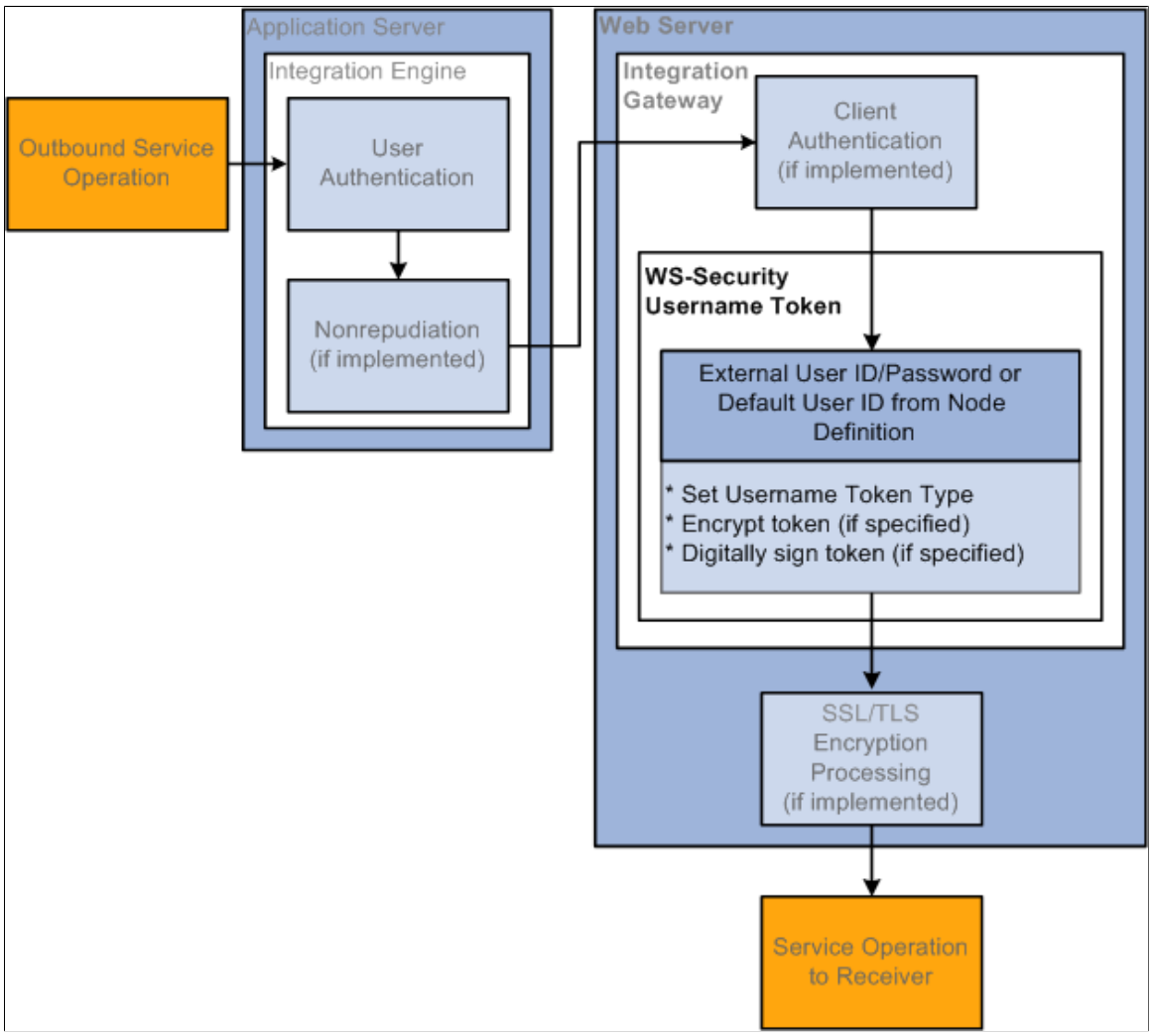
Outbound WS-Security Processing using Username Tokens

This section discusses outbound WS-Security processing using username tokens by PeopleSoft Integration Broker on outbound integrations.

WS-Security processing occurs on the integration gateway.

This diagram illustrates outbound WS-Security processing using username tokens when it is implemented. The diagram shows all possible security processing for an outbound integration to show where in the processing flow WS-Security processing occurs. WS-Security processing is highlighted in the foreground of the diagram.

The diagram illustrates the steps of WS-Security processing using username tokens, including validating the external user ID/password or default user ID from the node definition, the setting of the username token type, the encryption of the token if specified, and the digital signing of the token if specified.



When WS-Security is implemented for an outbound service operation, the integration gateway adds a WS-Security SOAP header to the service operation that contains UsernameToken credentials defined in the node definition for the node. The UsernameToken credentials can be comprised of any of the following from the node definition: *External User ID*, *External Password*, or *Default User ID*. Additionally, you can choose to encrypt and digitally sign the UsernameToken credentials.

See Implementing WS-Security for Outbound Integrations (Username and SAML Tokens), Describing WS-Security Configuration Options for Outbound Integrations (Username Tokens).

Understanding WS-Security Processing using SAML Tokens

This section provides overviews of:

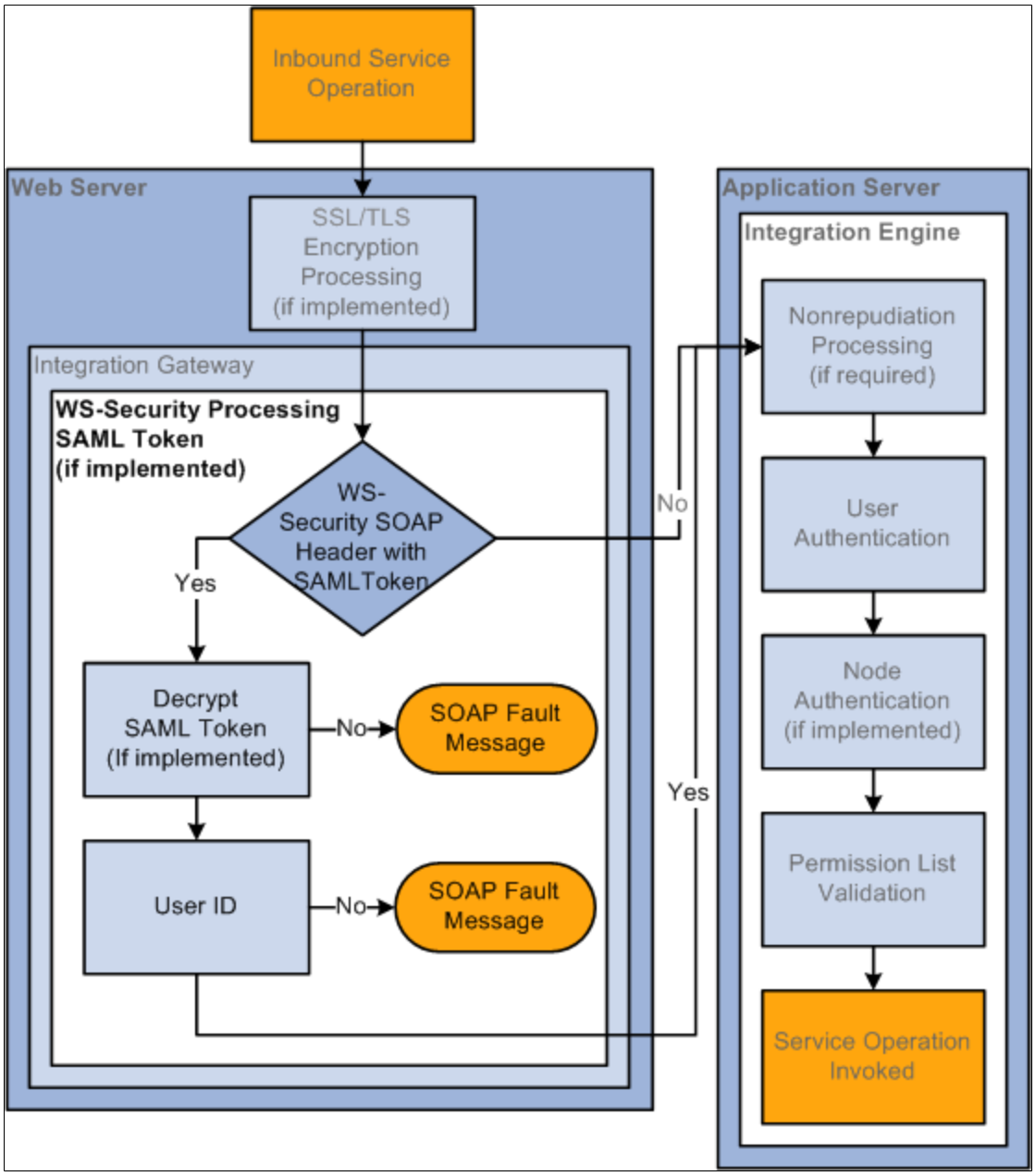
- Inbound WS-Security processing using SAML tokens.
- Outbound WS-Security processing using SAML tokens.

Inbound WS-Security Processing Using SAML Tokens

The inbound processing of service operations that are WS-Security-compliant using SAML tokens occurs on the integration gateway.

This diagram illustrates inbound WS-Security processing using SAML tokens when it is implemented. The diagram shows all possible security processing for an inbound integration to show where in the processing flow WS-Security processing occurs. WS-Security processing is highlighted in the foreground of the diagram.

The diagram illustrates the steps of WS-Security processing using SAML tokens, including validating the existence of a WS-Security SOAP header and SAML token, the decryption of the SAML token if encrypted, and the validation of the passed in user ID.



When any transaction arrives at the integration gateway, the PeopleSoft system checks for the existence of a WS-Security SOAP header. If it exists, the integration gateway decrypts the SAML token (if it has been encrypted) to restore the user ID information to clear text format.

The integration gateway then passes the user ID information to the application server, where additional security processing is performed.

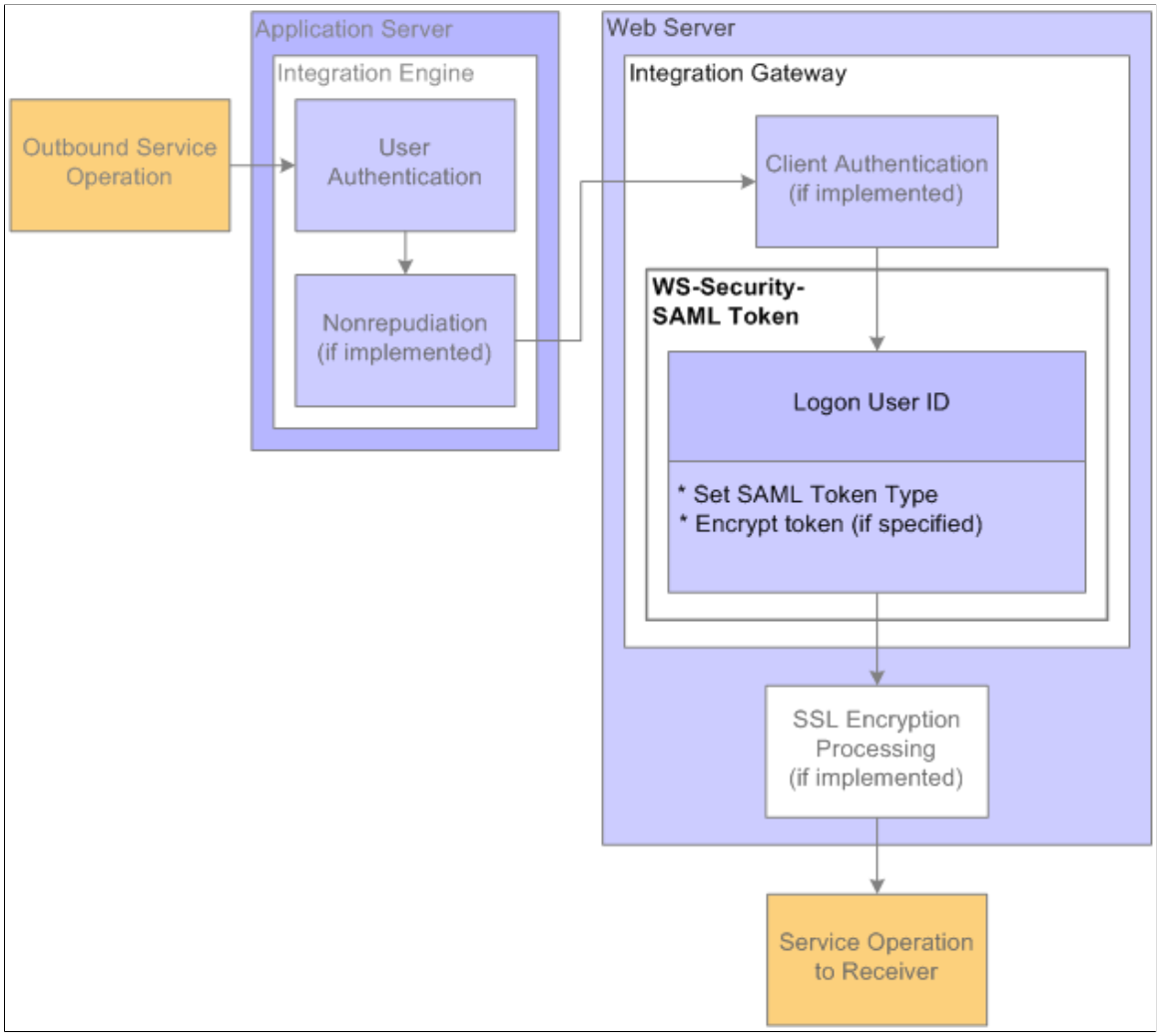
Outbound WS-Security Processing Using SAML Tokens

This section discusses outbound WS-Security processing using SAML tokens.

WS-Security processing occurs on the integration gateway.

This diagram illustrates outbound WS-Security processing using SAML tokens. The diagram shows all possible security processing for an outbound integration to show where in the processing flow WS-Security processing occurs. WS-Security processing is highlighted in the foreground of the diagram.

The diagram illustrates the steps of WS-Security processing using SAML tokens, including validating the logon user ID, the setting of the SAML token type, and finally the encryption of the token if encryption is specified.



When WS-Security is implemented for an outbound service operation, the integration gateway adds a WS-Security SOAP header to the service operation that contains SAML credentials defined in the node definition for the node. The SAML credentials can be comprised of any of the following from the node definition: *Default User ID* or *PeopleSoft/Single Signon User ID*. Additionally, you can choose to encrypt SAML credentials.

Warning! An any-to-local routing must be used on the sending system for outbound integrations using WS-Security processing and SAML tokens. Integrations will fail using any other routing type.

Prerequisites for Implementing WS-Security in PeopleSoft Integration Broker

To implement WS-Security in PeopleSoft Integration Broker you must install digital certificates.

It is also helpful to set the integration gateway logging to *5* as doing so enables you to see the WS-Security tags in the logs.

See Installing Integration Gateway-Based Digital Certificates"Managing Integration Gateway Message and Error Logging" (Integration Broker).

Implementing WS-Security for Inbound Integrations (Username Tokens)

There is no set up required for implementing WS-Security on inbound integrations. The integration gateway handles all inbound processing.

Implementing WS-Security for Inbound Integrations (SAML Tokens)

This section discusses how to:

- Set up the PeopleSoft system for handling SAML tokens.
- Set up and configure digital certificates.

Setting Up the PeopleSoft System for Handling SAML Tokens

There is some overlap of the steps to set up the PeopleSoft system to handle SAML tokens for integrations using PeopleSoft Integration with those for integrations using WSRP.

The following list describes the steps to set up the PeopleSoft system to handle SAML tokens. Some of the documentation that describes how to perform these steps is located elsewhere in the product documentation, as many of the same set-up steps are required to use SAML tokens with WSRP.

See "Configuring WS-Security for PeopleSoft as a WSRP Producer" (Portal Technology)

- Create the SAML administrator user ID.

  See "Creating the SAML Administrator" (Portal Technology).

- Set up application server and integration gateway digital certificates.

  See "Installing Application Server-Based Digital Certificates" (Security Administration) and Installing Integration Gateway-Based Digital Certificates.

- Set SAML assertion data.

  See "Configuring the SAML Inbound Setup" (Portal Technology).

Implementing WS-Security for Outbound Integrations (Username and SAML Tokens)

This section discusses how to:

- Specify an authentication token.
- Specify a default user ID.
- Specify an external user ID and password.

Specifying Authentication Tokens

Use the WS-Security page in the Nodes component (IB_NODE) to set up WS-Security for outbound integrations.

To access the WS-Security page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** and click the **WS Security** tab.

This example illustrates the Nodes – WS-Security page.

The previous example shows the WS Security page that appears by default. The options that appear on this page vary, depending on the authentication token type with which you are working.

To set up WS-Security for outbound integrations:

1. Select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions**.

   The Nodes search page appears.

2. Select the external remote node with which you are integrating.

   The Node Definitions page appears.

3. Click the **WS-Security** tab.

   The WS-Security page appears.

4. From the **Authentication Token Type** drop-down list box select an authentication type. The options are:

   - *SAML Token.*
   - *Username Token.*

5. To include additional security options, choose any of the following:

   Additional information about the possible configuration combinations using these options is discussed elsewhere in this section.

   See Describing WS-Security Configuration Options for Outbound Integrations (Username Tokens).

Field or Control	***Description***
Encrypt	(Optional.) When you check this box, an **Encryption Level** drop-down list box appears which allows you to choose the level of encryption.
Digitally Signed	This option appears only when you select *Username Token.* (Optional.) Check the box to digitally sign the token information, including the username and password.
Use Default User ID	This option appears only when you select SAML Token. (Optional.) Check the box to use the Default User ID specified on the Node Definitions page. If this option is not selected the user ID used is the PeopleSoft single signon user ID.
Use External User ID	This option appears only when you select Username Token. (Optional.) Check the box to use an external user ID for the username. If you select this option, you specify the external user ID and optional password (recommended) on the Node Definitions page. **Note:** If you do not select this option, the Default User ID specified on the Node Definition page is used as the username in the UsernameToken credential.

6. Click the **Save** button.
7. Click the Node Definitions tab.

The Node Definitions page appears.

If you chose to use an external user ID, a dialog box appears indicating that you need to specify the external user ID and optional password. Information on performing that task is described in the Specifying External User IDs and Passwords section.

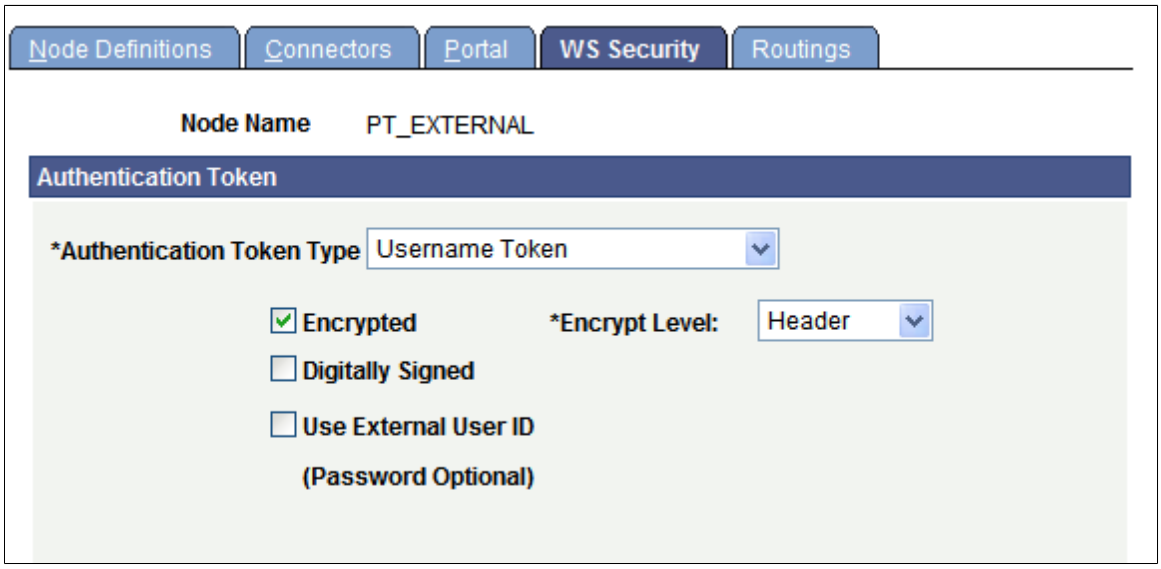
Encrypting Outbound Messages

When you choose to encrypt messages in an outbound service operation, you have the option to encrypt the entire message, the message body only, or the message header only.

Use the Nodes – WS Security page (IB_NODESECURITY) to work with any of these options.

To access the WS-Security page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions** and click the **WS Security** tab.

This example illustrates the Nodes – WS Security page. Use this page to choose a message encryption level.



When working with an external node type and you select the Encrypt box, the Nodes — WS Security page displays an Encrypt Level drop-down list box from which you can choose an encryption level.

The encryption level options are:

- *All.* This option encrypts the entire message including the message header and body.
- *Body.* Encrypts the message body only.
- *Header.* (Default) Encrypts the message header only.

Specifying Default User IDs

When using the SAML token type to implement WS-Security, you have the option to use the default user ID for processing. You set the default user ID on the external remote node definition using the Node Definitions page.

When you create a node definition, you must supply a value for the Default User ID. However, you a free to change the value at any time.

See Configuring Nodes.

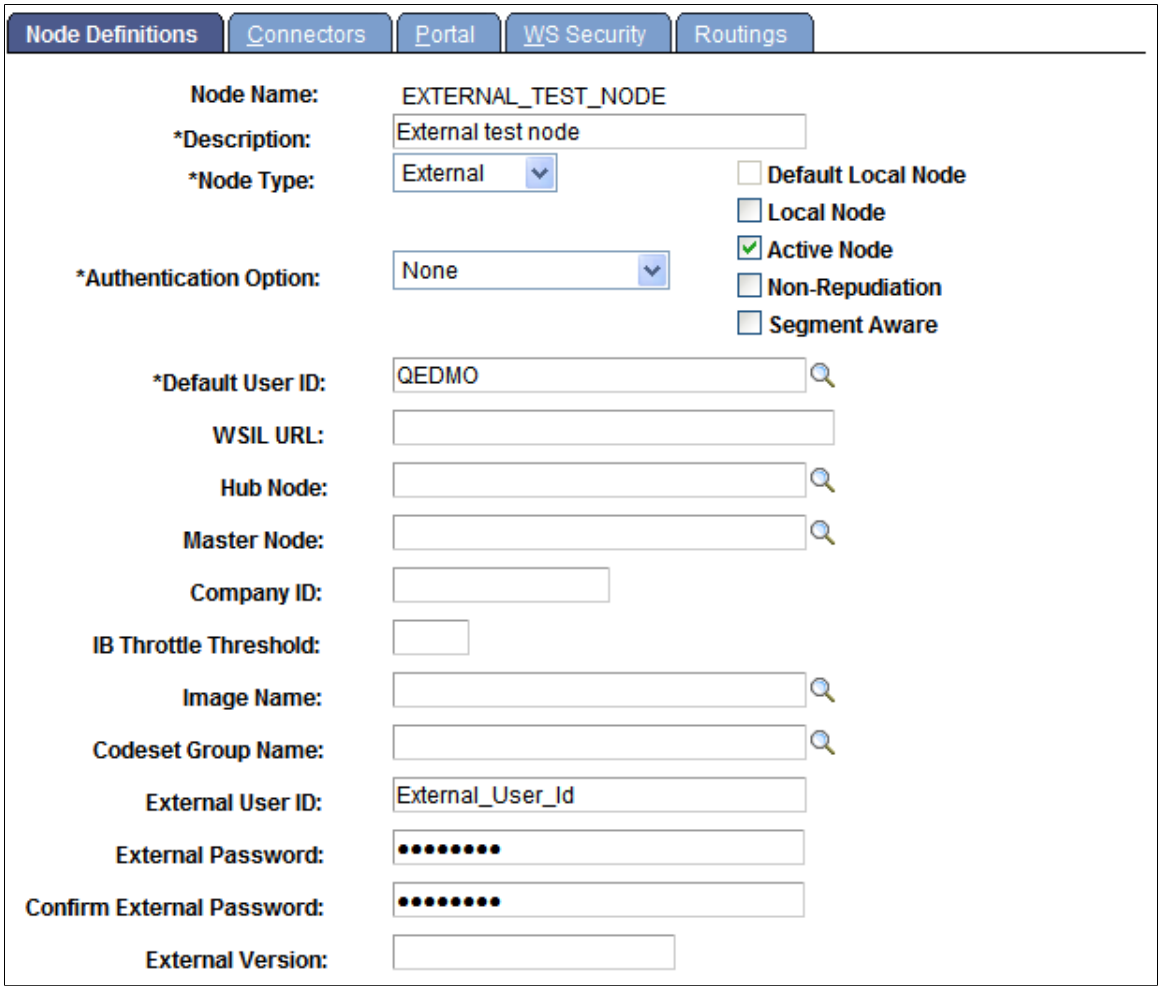
If you do not check the **Default User ID** option, the system uses the PeopleSoft User ID/Single Signon User ID for the transaction.

Specifying External User IDs and Passwords

When using the Username token type to implement WS-Security, you have the option to specify an external user ID. You define the external user ID on the Nodes – Node Definitions page.

To access the Nodes – Node Definitions page **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions.**

This example illustrates the Nodes – Node Definitions page.



When specifying an external user ID, specifying an external user ID password is recommended.

Note: The **Confirm External Password** field appears after you specify the external password and tab out of the field.

To specify the External User ID and Password:

1. On the Node Definitions page, in the **External User ID** field, enter an external user ID.
2. (Optional.) In the **External Password** field, enter the password for the external user ID.

   Tab out of the field. A **Confirm External Password** field appears.
3. In the **Confirm External Password** field, re-enter the external user ID password.
4. Click the **Save** button.

Development Considerations for Implementing WS-Security in Asynchronous Request/Response Service Operations

This section discusses development considerations for implementing WS-Security in asynchronous request/response service operations and discusses how to:

- Digitally sign responses in asynchronous request/response service operations.
- Secure responses in asynchronous request/response service operations.

Digitally Signing Responses in Asynchronous Request/Response Service Operations

This section applies to inbound asynchronous request/response service operations defined with any-to-local routing definitions.

In any-to-local routing definitions, no requesting node is present. As a result no digital certificate information that is normally defined at the node level is included with the request. However, the request does contain a RequestAliasName parameter that is populated with the certificate issuer's credentials that the integration gateway uses to process the request.

To digitally sign a response for an asynchronous request/response service operation, in the response set the RequestAliasName parameter equal to the same value that was set for the parameter on the request message. The integration gateway reads that value and can then determine the certificate to use in the response.

The following example shows how to code a digitally-signed response for an asynchronous request/response service operation:

```
import PS_PT:Integration:INotificationHandler;

class FLIGHTDATA_RETURN implements PS_PT:Integration:INotificationHandler
   method FLIGHTDATA_RETURN();
   method OnNotify(&MSG As Message);
end-class;

/* constructor */
method FLIGHTDATA_RETURN
end-method;

method OnNotify
   /+ &MSG as Message +/
   /+ Extends/implements PS_PT:Integration:INotificationHandler.OnNotify +/
   /* Variable Declaration */


   Local string &str, &value;
   Local Rowset &rs;
   Local integer &num;
```



```
   Local Message &MSG_resp;
   Local Record &FLIGHTDATA, &REC;

   &rs = &MSG.GetPartRowset(1);
   /* process request data */

   &MSG_resp = CreateMessage(Operation.FLIGHTPLAN_ARR, %IntBroker_Response);
   &rs = &MSG_resp.GetPartRowset(1);
   /* popualate response data */

   If &MSG.IsSourceNodeExternal Then

      /* set WS Addressing information and WS_RequestAliasName
   if security to be added to response message */

      &MSG_resp.IBInfo.WSA_MessageID = &MSG.IBInfo.WSA_MessageID;
      &MSG_resp.IBInfo.WSA_ReplyTo = &MSG.IBInfo.WSA_ReplyTo;
      &MSG_resp.IBInfo.WS_RequestAliasName = &MSG.IBInfo.WS_RequestAliasName;

   Else
      /* request from PSFT system */
      &MSG_resp.IBInfo.WSA_ReplyTo = &MSG.TransactionId;

   End-If;

   %IntBroker.Publish(&MSG_resp);
end-method;
```

Securing Responses in Asynchronous Request/Response Service Operations

PeopleSoft Integration Broker sends responses for asynchronous request/response service operations to the URL set in the Target Location field in on the Service Configuration page. The URL specified on this page is typically not secure, as it is the URL used for all WSDL, schemas, and web transactions.

When providing asynchronous request/responses, you can dynamically override the URL using the IBInfo object property WSA_ReplyTo. For example:

```
&MSG.IBINFO.WSA_ReplyTo
```

You set this property typically before the publish or during the OnSend event.

Overriding Node-Level WS-Security Settings on Routing Definitions

This section discusses how to:

- Override WS-Security settings on routing definitions (General).
- Override WS-Security settings on routing definitions (Synchronous Responses).
- Override WS-Security settings on routing definitions (Asynchronous Requests/Responses).

Understanding Overriding Node-Level WS-Security Settings on Routing Definitions

You can override node-level WS-Security settings on individual routing definitions for outbound request and response messages. The security settings that you can override are the same as those that appear on the Nodes-WS Security page.

When overriding WS-Security settings for synchronous request messages and asynchronous request/response messages, there are PeopleCode considerations of which you should be aware. In addition, the outbound security overrides on the routing definition for these transaction types work in concert with inbound security validation set on the service operation. These considerations and options are discussed in this section.

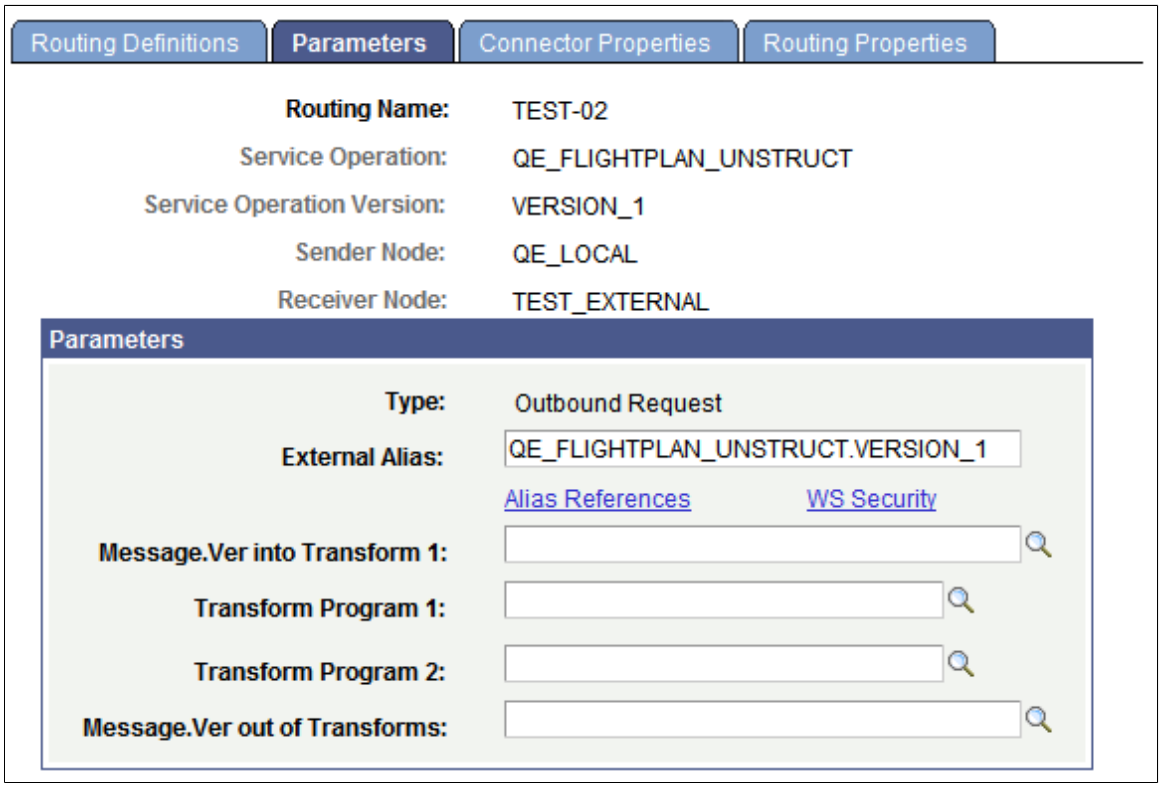
Overriding WS-Security Settings on Routing Definitions (General)

When you are working with an outbound routing definition to a external node, the Routing Definitions – Parameters page (IB_ROUTINGDEFNDOC) features a **WS-Security link** that provides access to options to override node-level WS-Security settings.

To access the Routing Definitions – Parameters page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Routing Definitions** and click the **Parameters** tab.

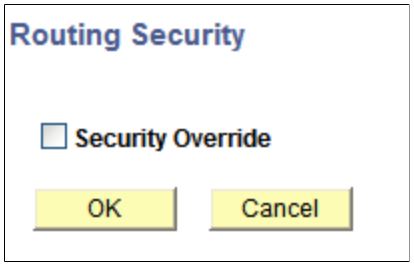
This example illustrates the Routing Definitions – Parameters page.

Use the **WS Security** link under the **External Alias** field to override node-level WS-Security settings.



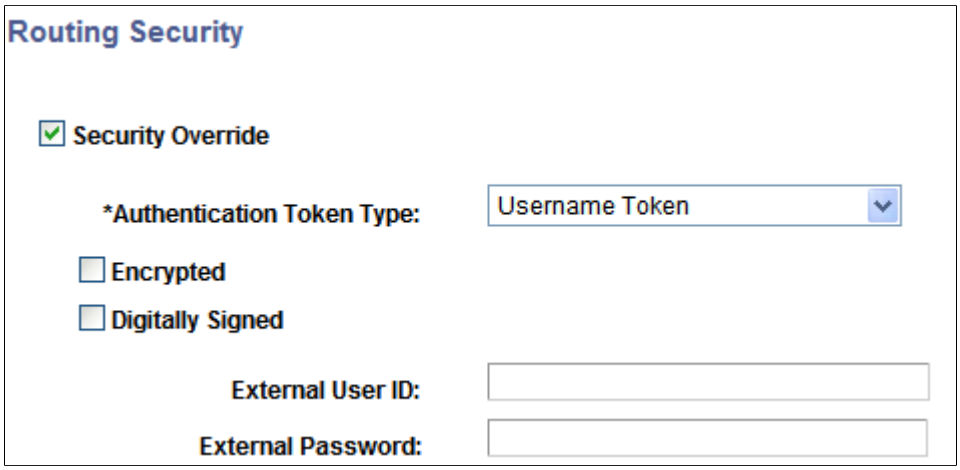
When you click the **WS Security** link the Routing Security page (IB_ROUTINGDEFN_SEC) appears.

This example illustrates the Routing Security page. The Routing Security does not display any override options until you check the **Security Override** box.



When you check the **Security Override** box on the Routing Security page, the WS-Security options that you can override appear on the page.

Check the Security Override button to view override options. This example illustrates override options when *Username Token* is the authentication type.



The WS-Security options that appear and that you can use to set and override in a routing definition, depend on the authentication type and encryption option, if any, set.

To override WS-Security options on a routing definition:

1. Select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Routing Definitions**, and select or add a routing definition.
2. Click the Parameters tab.

   A **WS Security** link appears for the outbound request or response message, depending on whether the external node is the sending or receiving node.
3. Click the **WS Security** link.

   The Routing Security page appears.
4. Select the **Security Override** box.

   The **Authentication Token Type** drop-down list box appears.
5. From the **Authentication Token Type** drop-down list box, choose an authentication token type with which to work. The options are:

- *None.* (Default.)
- *SAML Token.*
- *Username Token.*

If you choose SAML token or Username token, additional security options appear with which to work. SAML token and Username token security options for outbound messages are discussed elsewhere in this topic.

See Implementing WS-Security for Outbound Integrations (Username and SAML Tokens).

Overriding WS-Security Options on Routing Definitions (Synchronous Responses)

To override the security for a synchronous response, on the Routing Definitions – Parameters page, select the **WS Security** link on the Outbound Response Parameter:

If the **Encrypted** check box is selected then the response message sent from the consumer must be digitally signed if the routing is an any-to-local type routing.

In addition, you can reject a request message that is not digitally signed. To do so, on the Service Operations-General page, from the **Security Verification** \ drop-down list select Digitally Signed.

Overriding WS-Security Options on Routing Definitions (Asynchronous Request/Response)

You can use the Routing Security page to encrypt and/or digitally sign outbound responses in asynchronous request/response transactions.

If you select the **Encrypted** box for the outbound response message, then the message sent from the consumer must be digitally signed if the routing is an any-to-local type routing.

You can then select the *Digitally Signed* option for Security Verification on the service operation to reject a non-signed request message.

To successfully use WS-Security on a response, within the PeopleCode the *RequestAliasName* must also be populated on the response Message object from the request Message object. Here is an example in PeopleCode:

```
&MSG_resp = CreateMessage(Operation.FLIGHTPLAN_DOC_ARR, %IntBroker_Response);
&MSG_resp.IBInfo.WSA_MessageID = &MSG.IBInfo.WSA_MessageID;
&MSG_resp.IBInfo.WSA_ReplyTo = &MSG.IBInfo.WSA_ReplyTo;
&MSG_resp.IBInfo.WS_RequestAliasName = &MSG.IBInfo.WS_RequestAliasName;
```

The system validates proper encryption based on the request verification selected on the service operation. Select the desired security validation. If the security on the actual request message does not match the value specified in the Security Verification field, an error is sent back to the client.

Implementing WS-Security on Services Consumed Using the Consume Web Service Wizard

You can implement WS-Security on service operations you consume using the Consume Web Service wizard.

When using the Consume Services wizard, there is an option to use the default pre-defined *WSDL_NODE* node or use another existing node as the receiving node for the consumed service operations. The action to take to implement WS-Security on consumed services depends on which of these nodes you are using.

Using the Default WSDL Node

If you choose the default *WSDL_NODE* node as the receiving node, then you should add/override the node-level WS-Security settings by using the Routing Security page on the routing definition for the created service operation.

If you are using the *WSDL_NODE* node as the receiving node and the message is to be encrypted, the WS_RequestAliasName must be populated on the request message with the alias name used when adding the provider certificate to the gateway keystore. In addition, in PeopleCode after the message is created add this alias as follows:

```
&MSG.IBinfo.WS_RequestAliasName = "the alias name from the gateway keystore";
```

Using an Existing Node

If you use a node other than the *WSDL_NODE* as the receiving node then you can specify the security settings at the node level on the Nodes- WS Security page or on the Routing Security page on the routing definition.

Describing WS-Security Configuration Options for Outbound Integrations (Username Tokens)

This section discusses:

- Recommended WS-Security configurations for outbound integrations.
- Supported WS-Security configurations for outbound integrations.
- Non-secure WS-Security configurations for outbound integrations.

Recommended WS-Security Configurations for Outbound Integrations (Username Token)

The following table highlights recommended WS-Security configurations on the PeopleSoft system for outbound integrations using Username tokens. Note that the configuration is always performed on the remote node and that the node type is always *External*.

External User ID and Password	*Authentication Type*	*With SSL Encryption*	*Results*
Both	Username Token with the External User ID option.	Yes	The system uses the external user ID and password to generate the username token. The token is generated in clear text.
Both	Username Token with the following other options: • External User ID. • Encrypted. • Digitally signed.	No	The system uses the external user ID and password to generate the username token. The token is encrypted and digitally signed.
Both	Username Token with the Digitally signed option.	Yes	The system uses the external user ID and password to generate the username token. The token is digitally signed.
External User ID only.	Username Token with the External User ID option.	Yes	The system uses the external user ID to generate the username token. The token is generated in clear text.
External User ID only.	Username Token with the following other options: • External User ID. • Encrypted. • Digitally signed.	No	The system uses the external user ID to generate the username token. The token is encrypted and digitally signed.
External User ID only.	Username Token with the following other options: • External User ID. • Digitally signed.	No	The system uses the external user ID to generate the username token. The token is digitally signed.

Supported WS-Security Configurations for Outbound Integrations (Username Token)

The following table highlights supported WS-Security configurations on the PeopleSoft system for outbound integrations using Username tokens. Note that the configuration is always performed on the remote node and that the node type is always *External*.

External User ID and Password	***Authentication Type***	***With SSL Encryption***	***Results***
None.	Username Token option only.	Yes.	The system uses the default user ID defined on the node definition to generate the username token. The token is generated in clear text.
None.	Username Token with the following other options: • Encrypted. • Digitally signed.	No.	The system uses the default user ID defined on the node definition to generate the username token. The token is encrypted and digitally signed.
None	Username Token with the Digitally Signed option.	No.	The system uses the default user ID defined on the node definition to generate the username token. The token is digitally signed.

Non-Secure WS-Security Configurations for Outbound Integrations (Username Token)

The following table highlights non-secure WS-Security configurations on the PeopleSoft system for outbound integrations using Username tokens. Note that the configuration is always performed on the remote node and that the node type is always *External*.

Warning! The following configurations are not secure! This information is provided to advise you about configurations that can lead to breaches in security. Use the recommended or supported configurations discussed in the previous sections for configuring your system .

External User ID and Password	*Authentication Type*	*With SSL Encryption*	*Results*
Both	Username Token with the External user ID option.	No.	The system uses the external user ID and password to generate the username token. The token is generated in clear text.
None.	Username Token option only.	No.	The system uses the default user ID defined on the node definition to generate the username token. The token is generated in clear text.
Both	Username Token with the following options: • External user ID. • Encrypted.	No.	The system uses the external user ID and password to generate the username token. The token is encrypted.
None.	Username Token with the following options: • External user ID. • Encrypted.	No.	The system uses the default user ID and password to generate the username token. The token generated is encrypted.

WS-Security SOAP Header Examples (Username Token)

This section provides the following WS-Security code examples:

- WS-Security UsernameToken in ciphertext and digitally signed.
- WS-Security UsernameToken with clear text user name and password.
- WS-Security UsernameToken with clear text with user name only.

Example 1: WS-Security UsernameToken in Ciphertext and Digitally Signed

The following code example shows a WS-Security SOAP header that contains a UsernameToken in cipher text and that is digitally signed. This is the most secure configuration for WS-Security in PeopleSoft Integration Broker.

```
<soapenv:Header>
   <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.
     oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0
      .xsd">
      <xenc:EncryptedKey>
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/
```



```
      xmlenc#rsa-1_5"/>
   <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <wsse:SecurityTokenReference>
         <ds:X509Data>
            <ds:X509IssuerSerial>
               <ds:X509IssuerName>CN=PeopleTools TEST root CA,
                 DC=peoplesoft,DC=com,OU=PeopleTools Development,
                 O=PeopleSoft Inc,L=Pleasanton,ST=CA,C=US</ds:
                 X509IssuerName>
               <ds:X509SerialNumber>174697022083003580418117</ds:
                 X509SerialNumber>
            </ds:X509IssuerSerial>
         </ds:X509Data>
      </wsse:SecurityTokenReference>
   </ds:KeyInfo>
   <xenc:CipherData>
      <xenc:CipherValue>q8ytyn0kRisc3i7GwGtoQuU6NSXfvSNoJg76PWpppt
       4b4DoH8bRObvht8GLu904OExYBrNDB26qqOlKVpIzGrCJFgetlhikGghH/u2
       9GC96+YfFdxSFqcJo5PpJR1KnVZP0sKO4IHVIEcuxp7MonoV6dm5kd0d8atVw
       KXhJe5Yk=</xenc:CipherValue>
   </xenc:CipherData>
   <xenc:ReferenceList>
      <xenc:DataReference URI="#EncDataId-13925529"/>
   </xenc:ReferenceList>
</xenc:EncryptedKey>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
   <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/
       2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/
       xmldsig-more#rsa-sha256"/>
      <ds:Reference URI="#id-763474">
         <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/
             xml-exc-c14n#"/>
         </ds:Transforms>
         <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/
           xmlenc#sha256"/>
         <ds:DigestValue>cNBCuvnSP5MMlsJvaHMrZm9CsK0=</ds:
          DigestValue>
      </ds:Reference>
      <ds:Reference URI="#id-13925529">
         <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/
             xml-exc-c14n#"/>
         </ds:Transforms>
         <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/
           xmlenc#sha256"/>
         <ds:DigestValue>p+IodojBA2QzX6p9xe6PKJyUKSg=</ds:
          DigestValue>
      </ds:Reference>
   </ds:SignedInfo>
   <ds:SignatureValue>D/kTMJZvxnv7fjWzmvKC1xe8VSDiSz4lZDzFrf8q
     FFoXux+C2xD47TLWnD7m8ejp/Un3mzjWkVN8S4FpwRr/ymrxWTKWLrjCO
     zmjSW+ZbjGvs5UfpFyzEH7PWrXt+LnTeMKKJWYjzOi7HCHCVK9aC/RZCt
     7PkCbSZ7DJoOQO/lU=
   </ds:SignatureValue>
   <ds:KeyInfo Id="KeyId-28705465">
      <wsse:SecurityTokenReference wsu:Id="STRId-7131385" xmlns:wsu=
        "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
         wssecurity-utility-1.0.xsd">
         <ds:X509Data>
            <ds:X509IssuerSerial>
               <ds:X509IssuerName>CN=PeopleTools TEST root CA,DC=
                 peoplesoft,DC=com,OU=PeopleTools Development,
                  O=PeopleSoft Inc,L=Pleasanton,ST=CA,C=US
                </ds:X509IssuerName>
               <ds:X509SerialNumber>174332155640842765207620
               </ds:X509SerialNumber>
            </ds:X509IssuerSerial>
         </ds:X509Data>
```

```
            </wsse:SecurityTokenReference>
         </ds:KeyInfo>
      </ds:Signature>
      <xenc:EncryptedData Id="EncDataId-13925529" Type="http://www.w3.
        org/2001/04/xmlenc#Element">
         <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/
           xmlenc#tripledes-cbc"/>
         <xenc:CipherData>
            <xenc:CipherValue>wqrOr/efBcghEdcTPZMPqbrUu9mF+iCSLf2UhLYjOc
             Vg30+58TX3FCKXJhExi3iEdbuVrYt60mq3Maka6cg6+0JXw0Qmbjbl5qG8p
             sHajRtenvZc3dJeLRDclplbqUw65cDvBqQz+3+K5DBMh+tIlutf+0j3D9MiO
             3ht4Ni4bJ9Zk/h+DY9y05px2xtOMsXSrEhn4STGz4SdaOwFYHDUTT+y+D6zj
             GYxpRAexVQxAkjehW1JEGhyaqqdDmIYPJxCSy8JBc7xL/CaUng98ak8hAr38I
             obBt1qjlYjGo9VybfrX5j9lqn6pcrWX6x3o/9JYXeiaY36qHj+jVm0STq1fPr
             DDfh6ZI0/aeks83MnesMrX9bB7aKOo67DPjJstRvW/qfbIo3wYgv+3Jl68sHv
             u6p6GZEujaLIYIosJ+HtDzmZ2Q9aOtkk7+zFwDohkljAwmNSe3bt9e2i60pgF
             fVYcxg1Pwfz03MyKm83m5cLT9INb8LHK/GsKOl+9GvQ49nsJ6EYuAcPO4Q8Sr
             BvLVVPY3Qljw+4ZOZOEcndxVw+vU9n7cAMyeYa7p5Jpl6l2naeC4J98MIa16D
             CuVdvLIkipurkn2lbVYe5/m0SYbVibvTWE3BIQlWzF/mRHKkOhBhTaKg/Y/Q7
             sRlKcxKHtjnsjX2d4hTqTRYOoKFEH5sVi+gtyhgogiXRjg8wCAS68cYVwAFre
             W9xf2/ojGJFcO354Sk5rWt3GZzK8yRG5Jcgf5pgxnKC3LVgvvGPQM2Q/yGy1N
             OrXDhtzc80zM2SIOjv3A90Gzj9RGKzrWm+bw4QlhveY+rwyZGZRu3ibVUm+mi
             Ul7CdBBbrLOfz9xY45w3H2c6mtu98OwhuoiYHeVS/FkdpL+ztLmZi7gINIAQi
             sCZudpyKsZIcEhTPbTjQcdCVPZim1v9HFft00cSOE1u1CVEYNOSuCisrLJIch
             zAtE7gfa86/NcyEGmUBtvRsGVPkPq81cw1AosV8x4+KPCpTjxxeuMKGrowC2h
             Y/7DY+IYn4
            </xenc:CipherValue>
         </xenc:CipherData>
      </xenc:EncryptedData>
   </wsse:Security>
</soapenv:Header>
```

Example 2: WS-Security UsernameToken with Clear Text Username and Password

The following code example shows a WS-Security SOAP header that contains a clear-text UsernameToken credential that contains a user name and a password.

```
<soapenv:Header>
  <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
     <wsse:UsernameToken>
         <wsse:Username>IBUSER</wsse:Username>
         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
           oasis-200401-wss-username-token-profile-1.0#PasswordText">IBUSER1
         </wsse:Password>
     </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
```

Example 3: WS-Security UsernameToken with Clear Text User Name Only

The following code example shows a WS-Security SOAP header that contains a clear-text UsernameToken credential that contains only a user name. No password is specified.

```
<soapenv:Header>
  <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.
    oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
     <wsse:UsernameToken>
        <wsse:Username>QEMGR</wsse:Username>
        <wsse:Password/>
     </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
```


Implementing Nonrepudiation

This section provides and overview of nonrepudiation and discusses how to configure nonrepudiation.

Understanding Nonrepudiation

PeopleSoft Integration Broker applies nonrepudiation to cross-node messaging by digitally signing service operation requests and their responses.

Nonrepudiation Processing Overview

In PeopleSoft applications, nonrepudiation provides two-way protection; both the request and its response are nonrepudiated. PeopleSoft Integration Broker uses PKI technology to implement nonrepudiation for integrations. Each participating node's keystore contains its own private key and the public keys of the nodes with which it exchanges nonrepudiation service operations.

Nonrepudiation works in the following manner:

1. Node A generates a number, known as a *digest,* which uniquely identifies its service operation request.
2. Node A uses its private key to generate a signature based on the digest, and inserts the signature into the nonrepudiation service operation request.
3. Node A sends the nonrepudiation request to Node B.
4. When it receives the nonrepudiation request, Node B uses Node A's public key in its keystore to confirm the integrity of the digest.

   It then separately recreates the digest from the service operation, and compares it to the received digest to confirm the integrity of the service operation.
5. Node B generates a digest that uniquely identifies its response.
6. Node B uses its private key to generate a signature based on the digest, and it inserts the signature into the nonrepudiation response to confirm receipt of the nonrepudiation request.
7. Node B sends the nonrepudiation response to Node A.
8. When the nonrepudiation response is received, Node A uses Node B's public key in its keystore to confirm the integrity of the digest.

   It then separately re-creates the digest from the service operation and compares it to the received digest to confirm the integrity of the service operation content.

Nonrepudiation produces the following results:

- The sending node cannot repudiate that the service operation was sent, because the receiving node has a copy of the request signed by the sender.
- The receiving node cannot repudiate that the service operation was received and processed, because the sending node has a copy of the response signed by the receiver.

- The service operation integrity is verified, because the validated signature of each nonrepudiated service operation assures that the service operation content as received, exactly matches the content as sent.

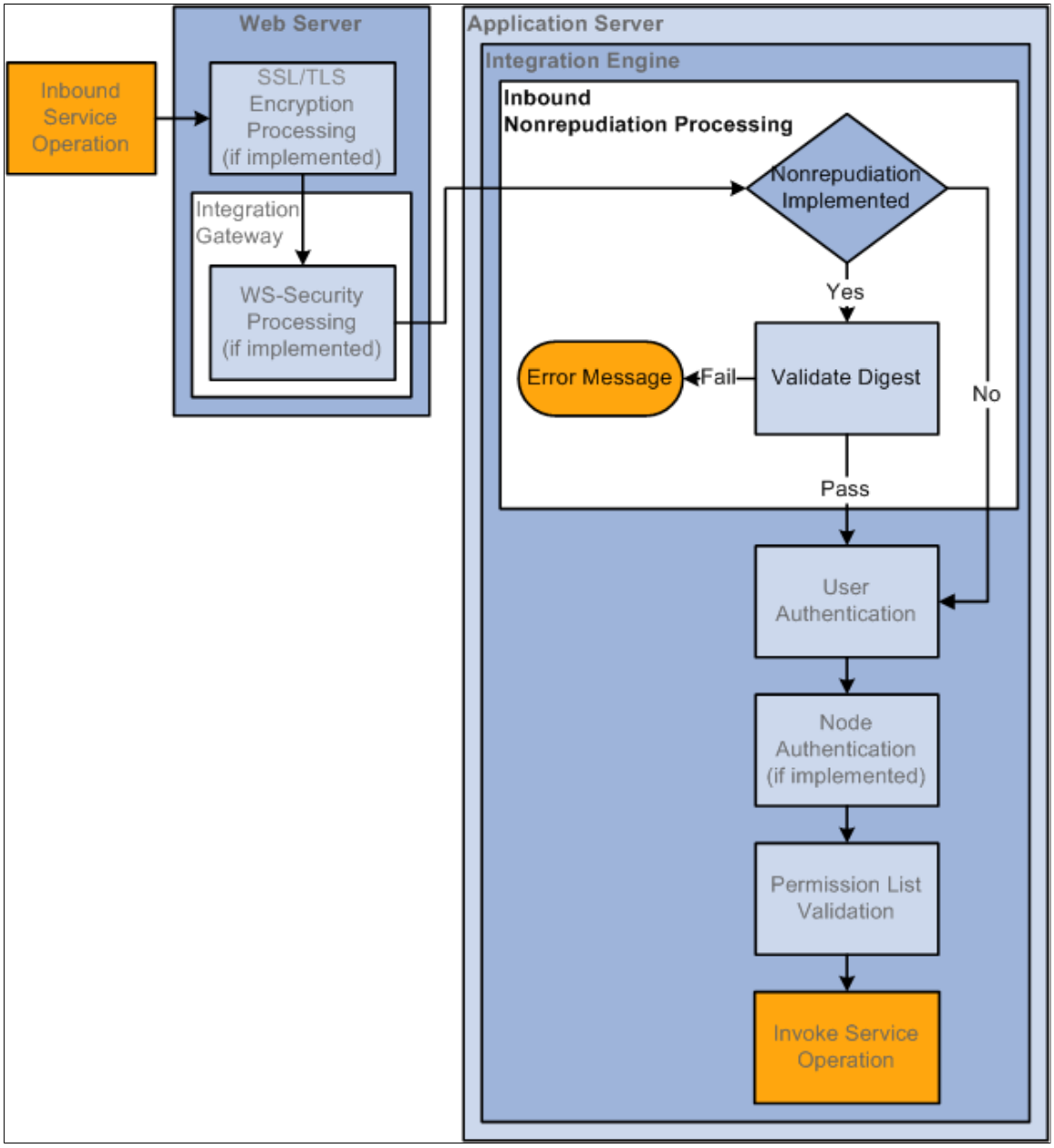
Inbound Nonrepudiation Processing

This section discusses inbound nonrepudiation processing.

Nonrepudiation processing occurs on the application server in the integration engine.

This diagram illustrates inbound nonrepudiation processing. The diagram shows all possible security processing for an inbound integration to show where in the processing flow nonrepudiation processing occurs. Nonrepudiation processing is highlighted in the foreground of the diagram.

The diagram illustrates the nonrepudiation processing steps on an inbound integration, including the validation that nonrepudiation is implemented, the validation of the nonrepudiation digest.



In inbound nonrepudiation processing, the system uses the integration partner's public key to validate the digest attached to the inbound service operation. It then uses its private key to recreate the digest on the service operation to validate the integrity of the service operation content.

If the system is able to validate the integrity of the digest and the service operation content, the service operation then goes through the user authentication process. If the system is unable to validate the digest or the service operation content, the transaction fails.

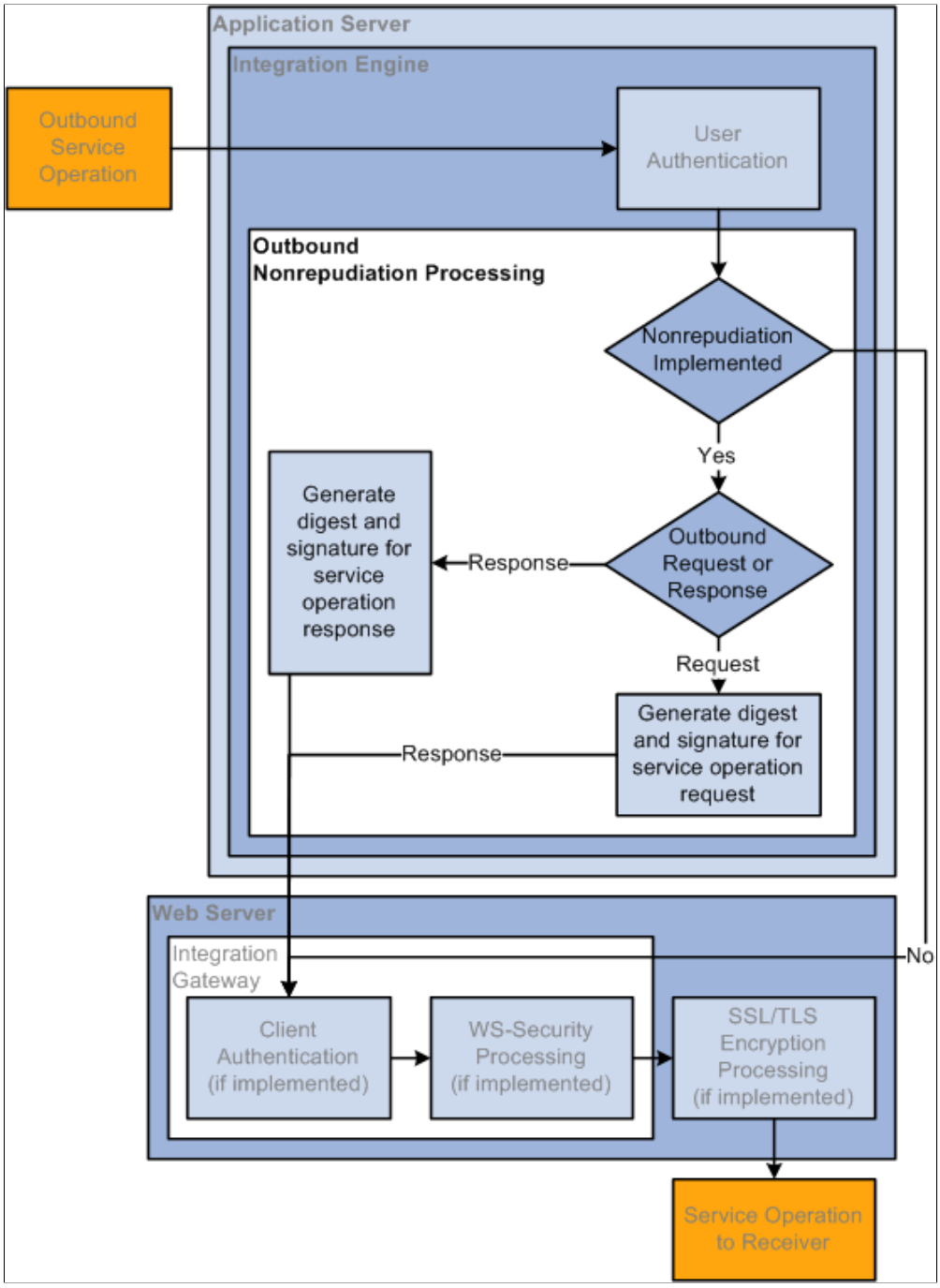
Outbound Nonrepudiation Processing

This section discusses outbound nonrepudiation processing.

Nonrepudiation processing occurs on the application server in the integration engine.

This diagram illustrates outbound nonrepudiation processing. The diagram shows all possible security processing for an outbound integration to show where in the processing flow nonrepudiation processing occurs. Nonrepudiation processing is highlighted in the foreground of the diagram.

The diagram illustrates the nonrepudiation processing steps on an outbound integration, including the validation that nonrepudiation is implemented, the determination if the message is a request or response, and then the generation of the digest and signature for the request or response.



On outbound service operations, the system determines if the service operation is a request or a response.

When the service operation is a request, the system uses its private key to generate a digest and signature, and attaches those items to the request.

When the service operation is an outbound response, the system uses its private key to generate a signature and response and inserts them into the service operation.

Prerequisites for Implementing Nonrepudiation

You must install application server-based digital certificates on both sending and target systems to implement nonrepudiation.

See “Installing Application Server-Based Digital Certificates” (Security Administration).

Configuring Nonrepudiation

This section discusses nonrepudiation configuration tasks on sending and target PeopleSoft systems using PeopleSoft Integration Broker.

If a participating node doesn’t use PeopleSoft Integration Broker, that node is still responsible for managing the appropriate private and public keys, inserting properly formatted signatures in the nonrepudiation service operation it sends, and properly handling signatures in the service operations that it receives.

With both archived and active nonrepudiation service operations, you can regenerate the digest in the Service Operations Monitor to reconfirm that it matches the attached digest.

See “Viewing Nonrepudiation Signatures in XML Format” (Integration Broker Service Operations Monitor).

Configuring Nonrepudiation on Sending PeopleSoft Systems

This section discusses configuring nonrepudiation on sending systems for asynchronous or synchronous transactions.

Prerequisites for configuring nonrepudiation are discussed elsewhere in this section.

See Prerequisites for Implementing Nonrepudiation.

To configure nonrepudiation for service operations on the source system you must:

- Select the **Non-Repudiation** check box on the service operation that will be invoked.
- Select the **Non-Repudiation** check box on the remote node definition that represents the target system.

Configuring Nonrepudiation on Target PeopleSoft Systems

You must supply the digital certificates containing the private and public keys required for nonrepudiation transactions.

No additional configuration is required on target PeopleSoft systems to handle nonrepudiated service operations. A nonrepudiated service operation received by a target PeopleSoft system will attempt to validate the service operation regardless if the local node and service operation are set for nonrepudiation.

Saving Nonrepudiated Service Operations

To save nonrepudiation service operations for future reference, you must archive them.

See “Archiving Service Operations” (Integration Broker Service Operations Monitor).

Managing User Authentication

This section provides overviews of user authentication, outbound user authentication, inbound user authentication, and discusses how to:

- Activate user authentication on service operations.
- Set up user authentication on sending systems.
- Exclude PeopleSoft authentication tokens in outbound requests to PeopleSoft nodes.

Understanding User Authentication

Access to invoke service operations is enforced at the user level.

When integrating with other PeopleSoft systems, user authentication determines the user ID to set on outbound integrations. The receiving system extracts this information and uses the user ID to validate against the permission list to which a service operation is assigned. If the user ID is assigned to the permission list, the sender can invoke the service operation.

When using Integration Broker for integrations among PeopleSoft systems, you must implement single signon and set up remote/target nodes as trusted nodes for user authentication to be validated. See “Implementing PeopleSoft-Only Single Signon” (Security Administration) for more information about implementing single signon and defining trusted nodes in the database.

Note: User authentication can be implemented on PeopleTools 8.48 and later systems only.

User IDs

The PeopleSoft system can use the following methods to set the user ID in an outbound transaction:

Term	*Definition*
Authentication Token	When the node is a PeopleSoft (*PIA*) node type, the PeopleSoft system automatically generates an authentication token and includes the token in the outbound transaction. The authentication token sets the user ID in the outbound transaction to the user ID that created the service operation.
Default User ID	The Node Definition page contains a *Default User ID* field. This is the user ID to which the node defaults, when no other user ID described in this section is set.

Term	*Definition*
External Name/External Password	You can programmatically set an external name and external password in the outbound SOAP message header or query string.
External User ID/Password	The Node Definitions page contains an *External User ID* and an *External Password* field. These fields are used in conjunction with WS-Security and are used for user authentication and to set the UsernameToken credentials for WS-Security processing. The *External Password* value is optional.

On inbound integrations from a PeopleSoft node, the PeopleSoft system looks for a user ID to associate with the permission list set for a service operation in the following order.

1. Authentication token.
2. Default User ID.

On inbound integrations not from a PeopleSoft node (External nodes and third-party systems), the PeopleSoft system looks for a user ID to associate with the permission list set for a service operation in the following order.

1. External Name/External Password.
2. External User ID/External Password.
3. Default User ID.

Understanding Outbound User Authentication

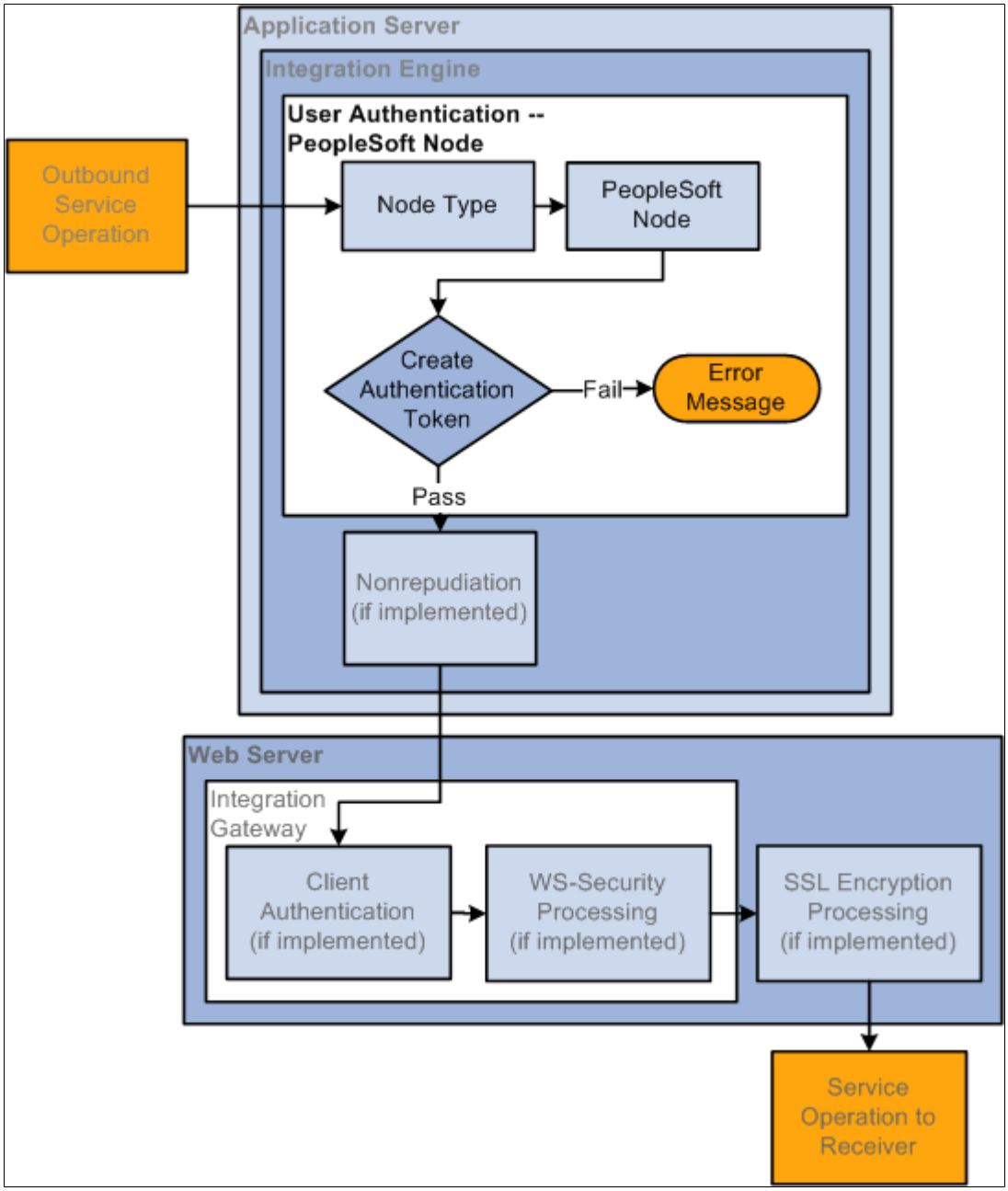
The outbound user authentication process determines the user ID to identify and attach to the outbound service operation. If the receiving system is a PeopleSoft system, the system validates the user ID and if the user ID belongs to the permission list to which the service operation is assigned, the service operation can be invoked.

The PeopleSoft system sets the user ID based on whether the sending node type is a PeopleSoft node (*PIA*) and by user ID information that may be defined in the SOAP message included with the service operation.

Outbound User Authentication: Sending Node is PeopleSoft Node Type

The following diagram illustrates the user authentication process when the local sending node is a PeopleSoft node:

The following diagram illustrates the user authentication process when the local sending node is a PeopleSoft node.



When the sending node is a PeopleSoft node, the user authentication process creates an authentication token to include in the transaction. The token is used on the receiving system to identify the sending node.

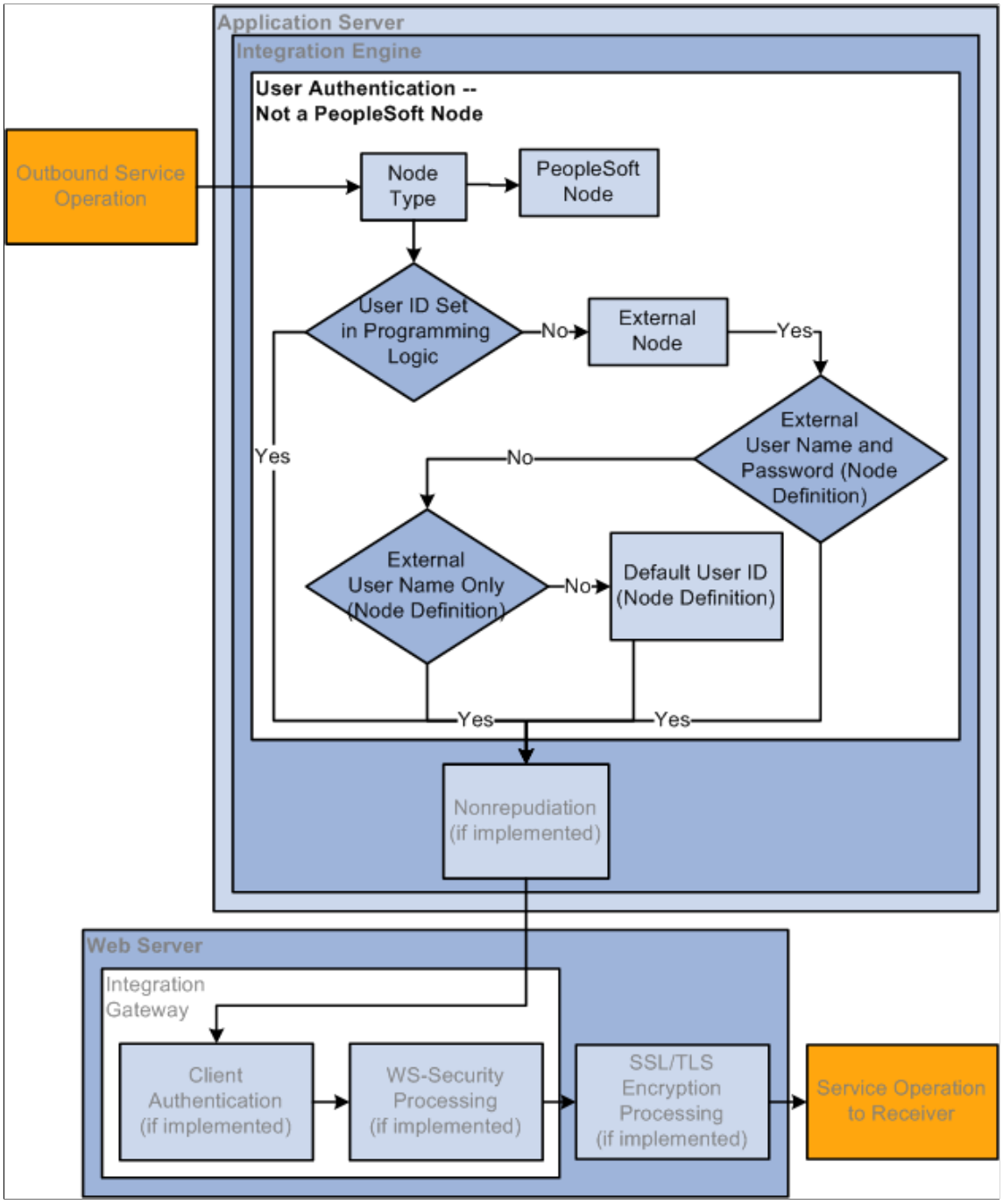
Note: The sending node must be defined as a trusted node on the receiving system for the PeopleSoft authentication token to be validated.

See Understanding User Authentication.

Outbound User Authentication: Sending Node is not PeopleSoft Node Type

The following diagram illustrates the user authentication process when the local sending node is not a PeopleSoft node type:

The following diagram illustrates the user authentication process when the local sending node is not a PeopleSoft node type.



When the sending node is not a PeopleSoft node, the system first looks at the SOAP message associated with the service operation to see if an external user ID or external user ID and password have been provided programmatically in the outbound SOAP message header. If so, the system uses that user ID/password and the service operation passes user authentication.

If an external user ID or external user ID and password are not specified programmatically in the SOAP message header, the system looks on the external node definition for user ID and password information. The system first looks for user ID and password information in the External User ID and External Password fields on the Node Definition page. If no External User ID or no External User ID/External Password is set, the system uses the Default User ID set on the Node Definitions page.

To summarize, when the sending node is not a PeopleSoft node type, the system follows this precedence for setting the user ID in the outbound service operation:

- User ID/password set in SOAP message header.
- User ID and password set in External User ID and External Password fields on the local external node definition.
- User ID set in the External User ID field on the local external node definition.
- User ID set in the Default User ID field on the local external node definition.

Understanding Inbound User Authentication

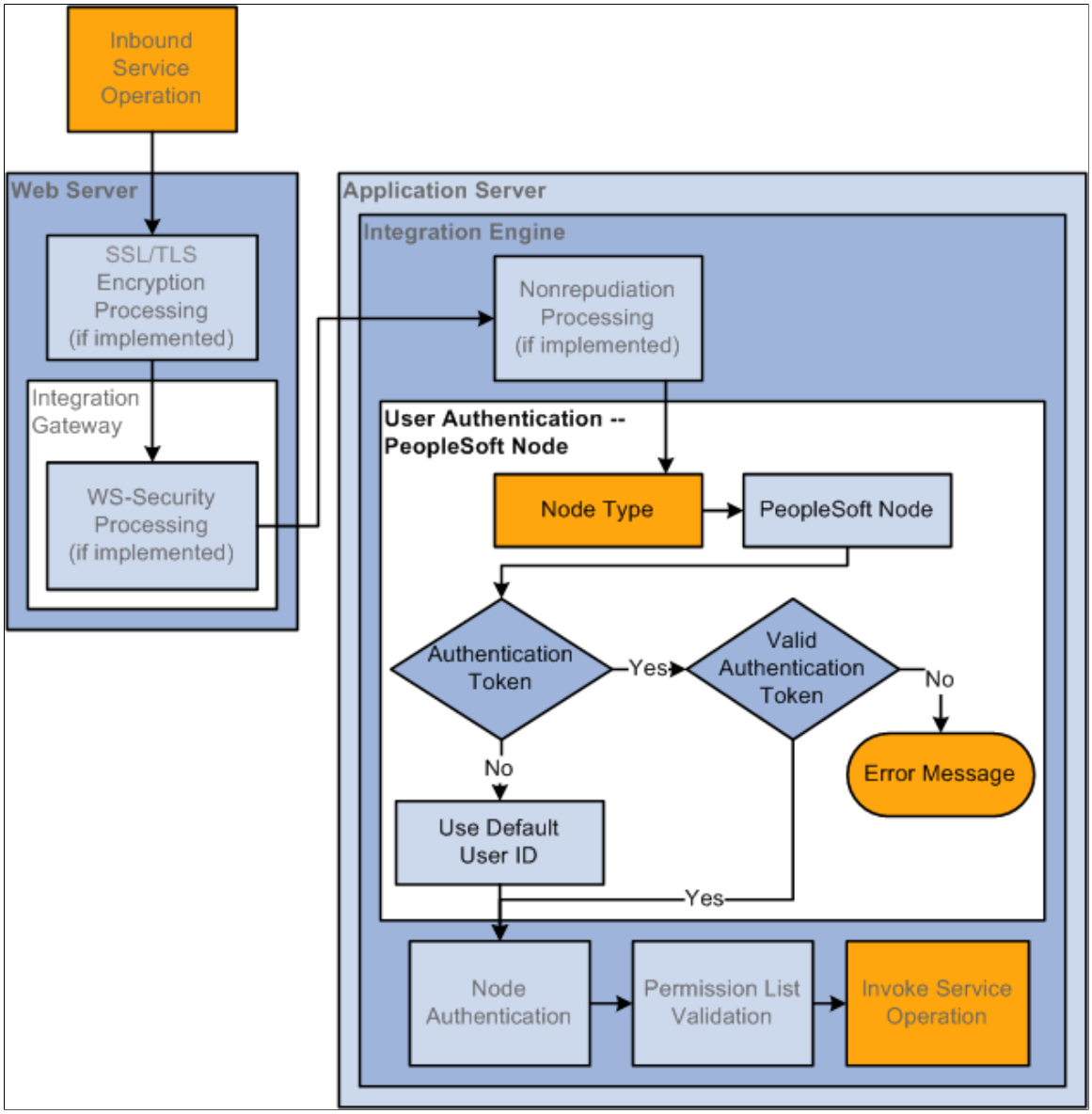
The inbound user authentication process determines the user ID that has been sent with an inbound service operation and determines if the sender is able to invoke the service operation.

The inbound user authentication process depends on whether the sender is a PeopleSoft node, the sender is an external node, or if the sender is not associated with any node. This section discuss user authentication processing for each of these situations.

Inbound User Authentication: PeopleSoft Node is the Sending Node

The following diagram illustrates the inbound user authentication process when a PeopleSoft node type is the sending node:

The following diagram illustrates the inbound user authentication process when a PeopleSoft node type is the sending node.



If the sending node is a PeopleSoft node, the system determines if an authentication token has been sent with the transaction. The system uses the authentication token to verify the sending node.

Note that the sending node does not need to be defined as trusted node on the receiving system for the PeopleSoft authentication token to be validated.

See Understanding User Authentication.

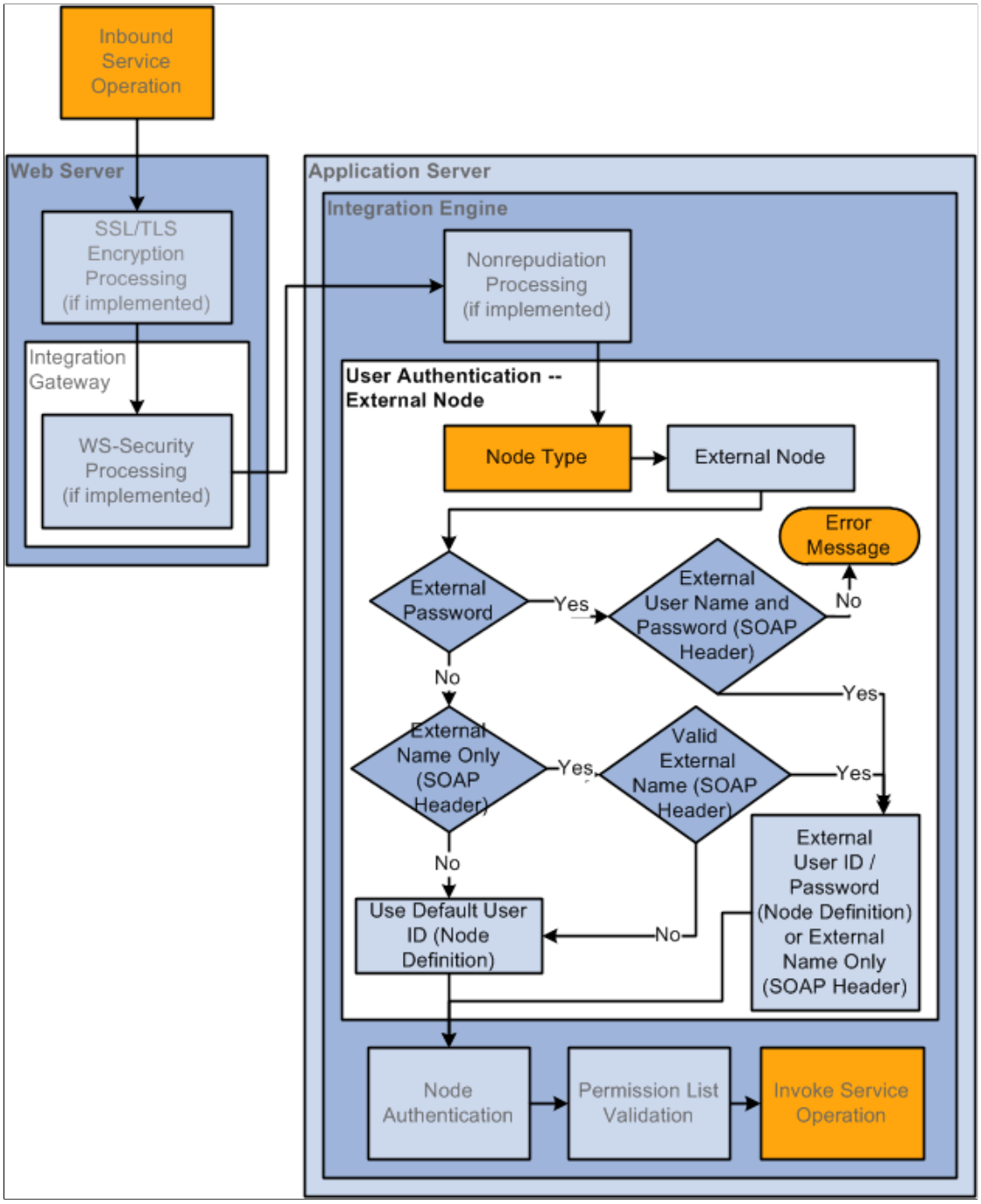
If authentication passes, the service operation has passed user authentication. If the authentication cannot be validated an error message is generated.

If no authentication token is included with the service operation, the system uses the default user ID on the external PeopleSoft node as the user ID.

Inbound User Authentication: External Node is the Sending Node

The following diagram illustrates user authentication processing when the sending node is an external node:

The following diagram illustrates user authentication processing when the sending node is an external node.



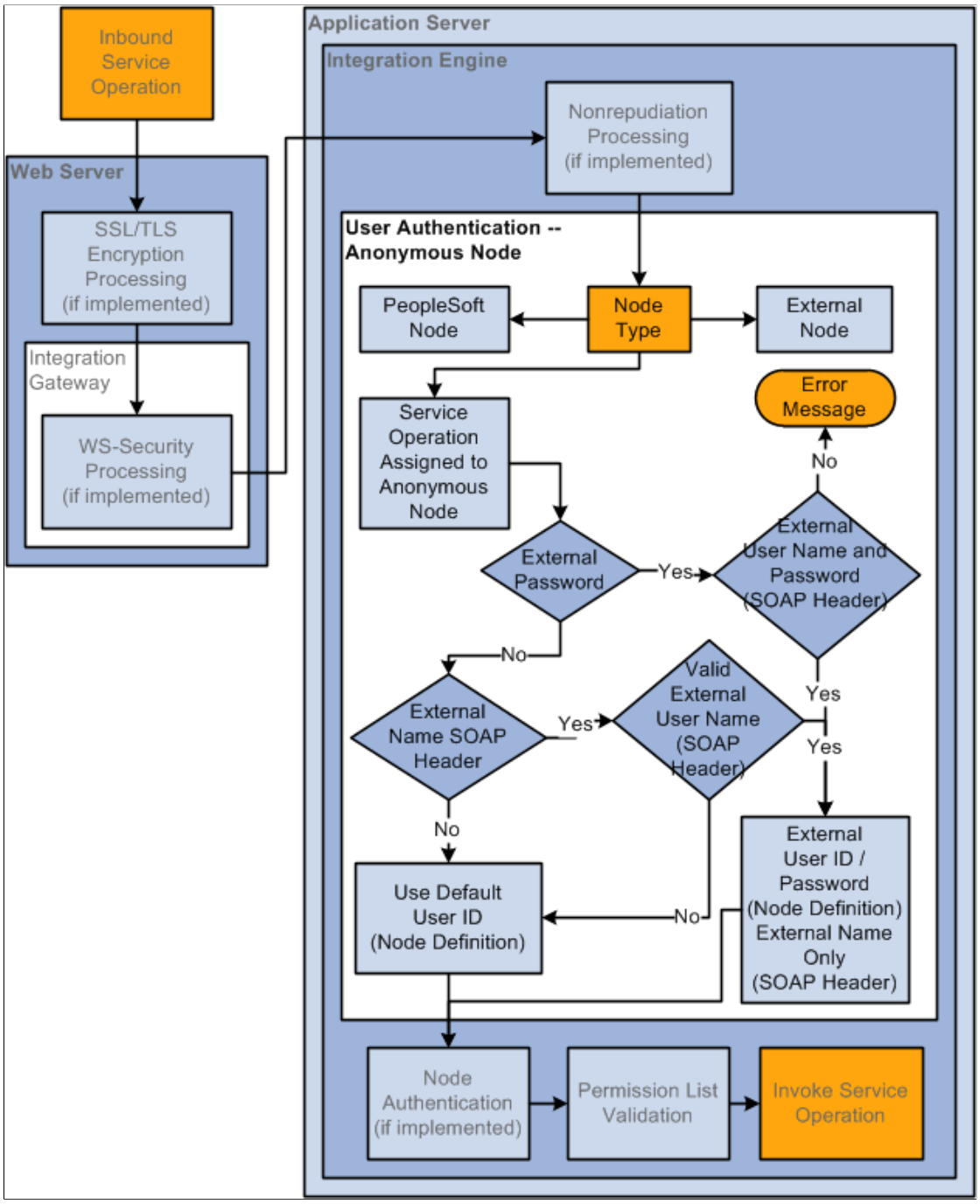
If the sending node is an external node type, the system first looks for a user ID and password set in the SOAP message header included with the inbound service operation. If both a user ID and password are not found, the system looks in the SOAP message header for a user ID only. If no user ID/password or

no user ID are found in the SOAP message header, the system uses the user ID set in the *Default User ID* field in the remote node definition.

Inbound User Authentication: Third–Party System Sending the Service Operation

The following diagram illustrates user authentication processing when a third-party system sends a service operation:

The following diagram illustrates user authentication processing when a third-party system sends a service operation.



Because third-party systems do not understand the concept of a node as defined and used within the context of PeopleSoft systems, PeopleSoft assigns transactions that have no node specified to a PeopleSoft-delivered Anonymous node.

If the PeopleSoft system first checks the SOAP message header for an external name and password set programmatically.

If none is found or if the system cannot validate the user ID or password that was set programmatically, it uses the *Default User ID* set on the Node Definitions page on the remote Anonymous node definition.

Activating User Authentication on Service Operations

To activate user authentication on a service operation:

1. Access the Service Operations-General page (**PeopleTools** > **Integration Broker** > **Integration Setup** > **Service Operation Definitions** and click the **General** tab.
2. Check the **User/Password Required** check box.
3. Save the changes.

Setting Up User Authentication on Sending Systems

This section discusses how to:

- Set up user authentication on remote PeopleSoft nodes.
- Set up user authentication on remote external nodes.
- Set up user authentication for third-party systems.

Understanding Setting Up User Authentication on Sending Systems

To set up user authentication on a sending system you must define the user ID on the remote node for the outbound transaction.

Setting Up User Authentication on Remote PeopleSoft Nodes

No set up is required to set up user authentication on a remote PeopleSoft (*PIA*) node type. An authentication token is automatically included in the outbound transaction. If the receiving system fails to authenticate the token an error message is returned. .

Setting Up User Authentication on Remote External Nodes

You can set the user ID for user authentication in any of the following ways on an external node:

- *External Name/Password*. Set programmatically in the SOAP message header or query string.
- *External User ID* and *External Password*. Set using the Node Definitions page.
- *Default User ID*. Set on the Node Definitions page.

Note: The user ID you specify must have access to the permission list to which a service operation is assigned to invoke the operation on the receiving system.

To access the Node Definitions page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Node Definitions.**

Setting Up User Authentication for Third-Party Systems

As discussed previously in this section, all inbound transactions that do not have PeopleSoft (*PIA*) node or external (*External*) node type specified are assigned to an Anonymous node.

You can set the user ID in requests from third-party systems programmatically in the external name/password elements in the outbound SOAP message header.

If the system does not find an external name or password in these locations, it uses the **Default User ID** field that you define on the remote Anonymous node.

Related Links

Defining Node Parameters

Excluding PeopleSoft Authentication Tokens in Outbound Requests to PeopleSoft Nodes

This section discuss how to exclude PeopleSoft authentication tokens in outbound requests to PeopleSoft nodes.

Understanding Excluding PeopleSoft Authentication Tokens in Outbound Requests to PeopleSoft Nodes

A PeopleSoft authentication token in an outbound request to a PeopleSoft target node signifies to the target PeopleSoft target system that the sender is a valid user on its system.

However, for some integrations there can be many users or validating users may not be warranted. In such cases you can exclude the PeopleSoft authentication token from inclusion in outbound requests to PeopleSoft target nodes.

When the PeopleSoft authentication token is excluded in a request, the default user ID for the sending node on the target system is the user ID used for integration authentication.

When you exclude PeopleSoft authentication tokens, user authentication is not performed. In lieu of user authentication, you can perform node authentication when authentication tokens are excluded. The following set-up is required on sending and receiving systems for node authentication:

- Install application server-based digital certificates.

  See “Installing Application Server-Based Digital Certificates” (Security Administration)

- Implement two-way SSL.

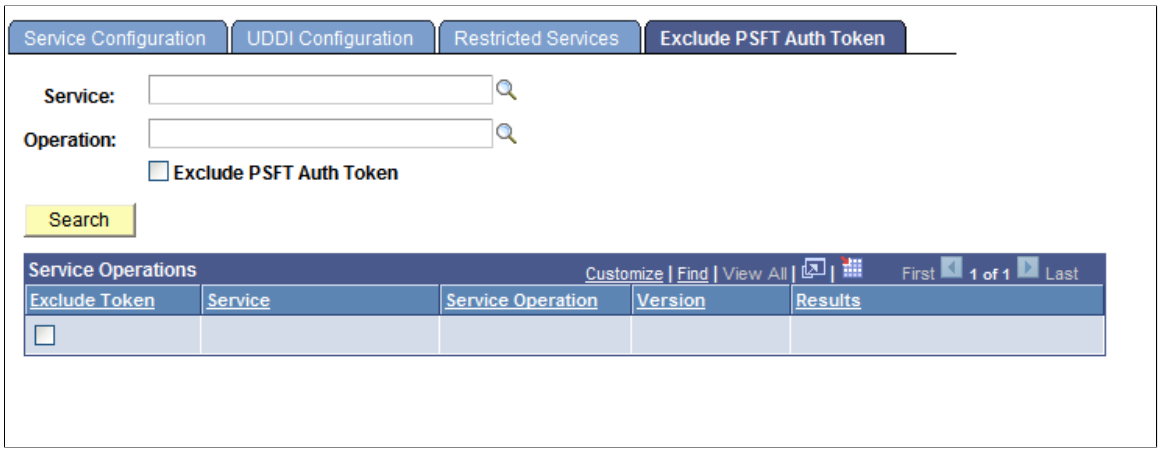
  See “Installing Web Server-Based Digital Certificates” (Security Administration)

Viewing Service Operations where PeopleSoft Authentication Tokens Have Been Excluded

Use the Exclude PSFT Auth Token page (IB_SVCSETUP5) to view service operations where PeopleSoft authentication tokens have been excluded.

To access the page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the Exclude PSFT Auth Token tab.

This example illustrates the Services Configuration – Exclude PSFT Auth Token page.



To view service operation where PeopleSoft authentication tokens have been excluded:

1. Access the Exclude PSFT Auth Token page (**PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the Exclude PSFT Auth Token tab).
2. Select the **Exclude PSFT Auth Token** box under the **Operation** field.
3. Click the **Search** button.

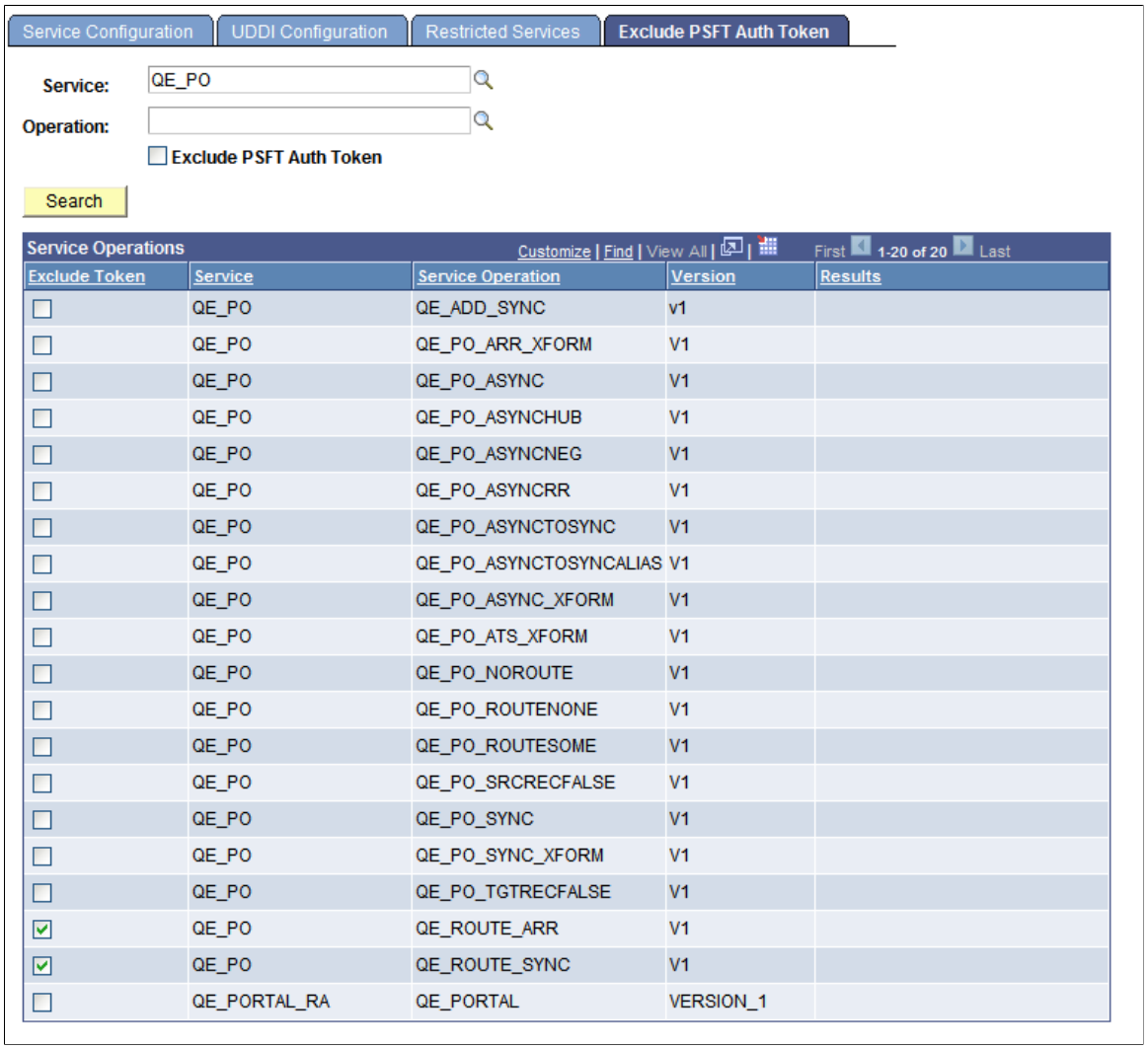
The system displays all service operations where the PeopleSoft authentication token has been excluded and will not be included in the service operation transaction.

Excluding PeopleSoft Authentication Tokens in Outbound Requests

Use the Exclude PSFT Auth Token page to exclude authentication tokens in outbound requests:

To access the page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the **Exclude PSFT Auth Token** tab.

This example illustrates the Services Configuration – Exclude PSFT Auth Token page. The example shows that the PeopleSoft authentication token has been excluded from the *QE_ROUTE_ARR* and *QE_ROUTE_SYNC* service operations



In the example shown, a search was performed on the service *QE_PO*. The *QE_ROUTE_ARR* and *QE_ROUTE_SYNC* service operations have been selected, and therefore the PeopleSoft authentication token will be excluded from those service operations. Scrolling to the right would reveal a Results column that indicates the selection was successful.

To exclude a PeopleSoft authentication token in an outbound request:

1. Access the Exclude PSFT Auth Token page (**PeopleTools** > **Integration Broker** > **Configuration** > **Service Configuration** and click the Exclude PSFT Auth Token tab).

2. Select one or more service operations from which to exclude the PeopleSoft authentication token:

    - To select one service operation, click the **Service** and **Operation** lookup buttons to locate the service operation. Click the **Exclude PSFT Auth Token** box.

    - To select multiple service operations, enter all or part of the service name or service operation name. Click the **Search** button. A list of results displays in the Service Operations section. Check

the **Exclude Token** box next to each service operation that should not include a PeopleSoft authentication token.

Note that you can also click the **Search** button to display all service operations in the database.

3. Click the **Save** button.

Implementing Node Authentication

This section discusses how to:

- Set up password-based node authentication.
- Set up certificate-based node authentication.

Understanding Node Authentication

You can implement node authentication with a password or digital certificates.

Setting Up Password-Based Node Authentication

To implement password authentication, you select the *Password* option from the **Authentication** drop-down list in a node definition, and enter a password. When you do this for a default local node definition, you must enter the same password in any remote node definition that represents the same node on the other participating systems.

See Defining Node Parameters.

Setting Up Certificate-Based Node Authentication

Certificate-based node authentication involves the following tasks:

- You must supply the digital certificates containing the private and public keys required for authenticated transactions.

  These elements are required at every node that participates in an authenticated transaction; PeopleSoft Integration Broker handles the mechanics of applying the keys.

- You must select the *Certificate* option from the **Authentication** drop-down list in a node definition.

  When you do this in a default local node definition, you must select the same option in any remote node definition that represents the same node on the other participating systems.

Related Links

“Installing Application Server-Based Digital Certificates” (Security Administration)
Defining Node Parameters

Securing Service Operations with Permission Lists

Securing Service Operations with Permission Lists is discussed here “Setting Permissions to Service Operations” (Integration Broker).

Validating Security on Inbound Integrations

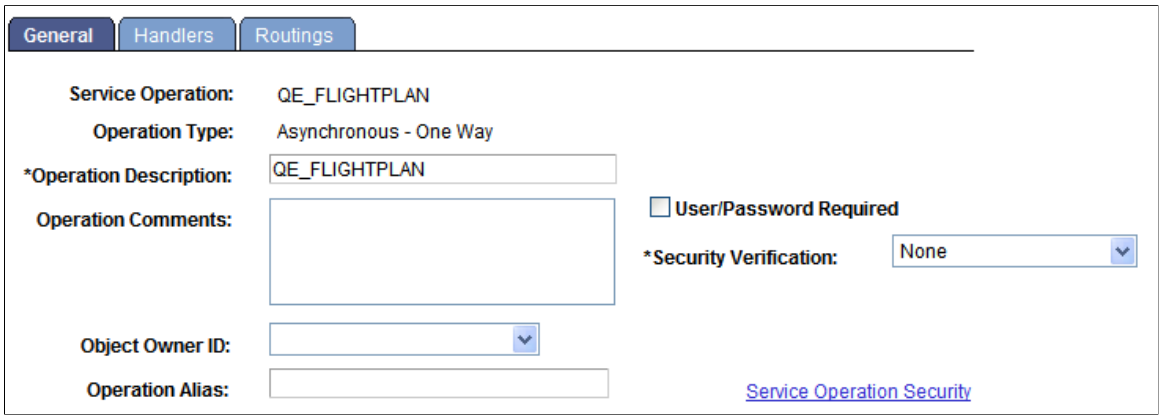
PeopleSoft Integration Broker can validate that inbound service operations from integration partners are transmitted with a level of security as determined by your organization. If they do not pass validation based on the parameters you set, the integrations are rejected.

The Service Operations – General tab (IB_SERVICE) features a **Security Verification** drop-down list box that enables you to set the required level of security on inbound integrations. For REST service operations the field is labeled **Req Verification.**

To access the Service Operations – General tab select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Service Operation Definitions.**

Validating Security on Inbound Service Operations

This example illustrates the portion of the Service Operations page for a non-REST service operation that contains the **Security Verification** drop-down list box. The **Security Verification** drop-down list box is located under the **User/Password Required** box.



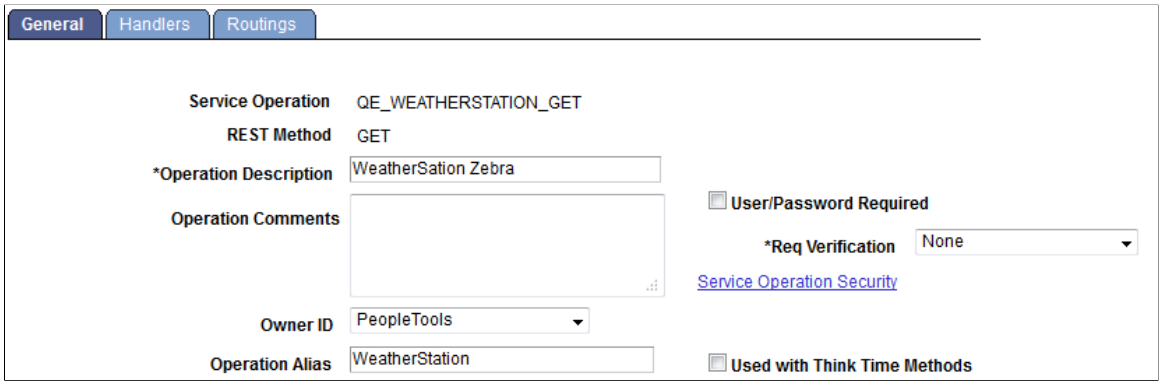
To require security on inbound (non-REST) service operations, select a value from the **Security Verification** drop-down list. The valid values are:

- *Digital Sign or SSL.* The integration partner must digitally sign the service operation or transmit it using SSL encryption.
- *Digitally Sign.* The integration partner must digitally sign the service operation.
- *Encrypt.* The integration partner must employ WS-Security encryption on the service operation.
- *Encrypt and Digitally Sign.* The integration partner must employ WS-Security encryption and digitally sign the service operation.

- *Encrypt or SSL.* The integration partner must employ WS-Security encryption on the service operation the service operation or transmit it using SSL encryption.
- *Encrypt/Digitally Sign or SSL.* The integration partner must employ WS-Security encryption and digitally sign the service operation, or transmit it using SSL encryption.
- *None.* (Default.) The integration partner is not required to set any specific security options on the service operation.
- *SSL.* The integration partner must transmit the service operation using SSL encryption.

Validating Security on Inbound REST Service Operations

This example illustrates the portion of the Service Operations page for a REST service operation that contains the **Req Verification** drop-down list box. The **Req Verification** drop-down list box is located under the **User/Password Required** box.



To require security on inbound REST service operations, select a value from the **Req Verification** drop-down list. The valid values are:

- *Basic Authentication.* The integration partner must pass the external user ID and password defined on the local node.
- *Basic Authentication and SSL.* The integration partner must pass the external user ID and password defined on the local node, and transmit the service operation using SSL encryption.
- *None.* (Default.) The integration partner is not required to set any specific security options on the service operation.
- *PeopleSoft Token.* The integration partner must pass a valid PeopleSoft Token.
- *PeopleSoft Token and SSL.* The integration partner must pass a valid PeopleSoft Token and transmit the service operation using SSL encryption.
- *SSL.* The integration partner must transmit the service operation using SSL encryption.
- *OAuth2.* The integration partner must pass a valid OAuth2 Token.
- *OAuth2 and SSL.* The integration partner must pass a valid and transmit the service operation using SSL encryption.

Related Links

“Installing Web Server-Based Digital Certificates” (Security Administration)
Implementing Web Services Security

Chapter 15

Tuning Messaging System Performance

Understanding Tuning Messaging System Performance

This topic discusses actions you can take to tune messaging system performance. This topic also features code examples to help illustrate concepts and features for tuning messaging system performance.

Note: The code examples in this topic are for illustrative purposes only and are not intended to be used in a production environment.

In addition, you can view messaging system performance statistics using the Service Operations Monitor.

Related Links

"Understanding Messaging System Performance Statistics" (Integration Broker Service Operations Monitor)

Throttling Dispatched Messages Through the Messaging System

You can throttle the number of dispatched messages from a given dispatcher to its associated handler(s).

Throttling the messages that pass through the messaging system enables you to avoid Tuxedo queue saturation due to redundant Tuxedo calls, which result in degraded performance.

You can throttle messages on any of the three pub/sub dispatchers:

- PSBRKDSP
- PSPUBDSP
- PSSUBDSP

To set up dispatcher throttling, you must set the following parameters located in PSADMIN:

- Tuxedo Queue Status Check Count
- Dispatcher List Multiplier
- Dispatcher Queue Max Queue Size

Information for setting these parameters is described earlier in this PeopleBook.

See Specifying Dispatcher Parameters.

Using Multi-Threading to Send Groups of Messages in Parallel

This section provides an overview of multi-threading and discusses how to:

- Specify the number of available threads.
- Implement multi-threading.

Understanding Multi-Threading

Multi-threading allows you to send a group of synchronous requests in parallel, thereby eliminating the need to wait for a response for one synchronous message to be returned before you send the next synchronous message. You can also use multi-threading to send a configurable number of asynchronous message publications in parallel.

Multi-threading enables you to pool request messages into an array and make a threaded call.

When working with synchronous messages, responses are returned in an array, and are pooled in the same order in which you send them.

Multi-threading supports sender-specified routing, thereby enabling you to pass in an array of nodes on the SyncRequest call.

Related Links

Implementing Exception Handling for Synchronous Message Processing

Specifying the Number of Available Threads

The number of threads available determines the number of message you can send in parallel. For example, if there are 10 threads available, you can send 10 messages in parallel.

To specify the number of threads available for multi-threading set the Thread Pool Size parameter in PSADMIN.

The thread pool size only affects the number of messages processed at the same time, and does not limit the number of messages you can send in one API call.

Setting the Thread Pool Size for Synchronous Messaging

For synchronous messaging, set the Thread Pool Size parameter in the General Settings for Integration Broker section in PSADMIN.

For synchronous messaging, The default value is *5*. The minimum value is *1* and the maximum value is *20*.

Setting the Thread Pool Size for Asynchronous Messaging

For asynchronous messaging, set the Thread Pool Size parameter in the Settings for Publication Contract Handler section in PSADMIN.

For asynchronous messaging, The default value is *1*. The minimum value is *1* and the maximum value is *20*.

Implementing Multi-Threading

This section provides the syntax for multi-threading and provides a synchronous multi-threading code example.

Syntax

The syntax for implementing multi-threading is:

```
Array of messages = %IntBroker.SyncRequest(Array of messages, array of
sender-specified routing);
```

The IntBroker object is responsible for managing the messages, instantiation of the SyncRequest handler and calling the Send method for each request. The IntBroker object then polls the SyncRequest handler object to determine when all processing is complete. At that time, status and error checking is performed and the response message objects are created. The response messages are packaged as an array and returned to the calling method.

Synchronous Multi-Threading Example

The following example shows code for synchronous multi-threading

```
Local Rowset &FLIGHTPLAN, &FLIGHTPLAN_RETURN;
Local Message &MSG;

Local array of Message &messages;
Local array of Message &return_mesages;

&messages = CreateArrayRept(&MSG, 2);
&return_mesages = CreateArrayRept(&MSG, 2);

&FLIGHT_PROFILE = GetLevel0();
&messages [1] = CreateMessage(Message.QE_FLIGHTPLAN_SYNC);
// populate the rowset
&messages [1].CopyRowset(&FLIGHT_PROFILE);

&messages [2] = CreateMessage(Message.QE_FLIGHTPLAN_SYNC);
// populate the rowset
&messages [2].CopyRowsetDelta(&FLIGHT_PROFILE);

&return_mesages = %IntBroker.SyncRequest(&messages);

// process the return rowset
&FLIGHTPLAN_RETURN = &return_mesages [1].GetRowset();
&temp = &return_mesages [1].GenXMLString();

// process the return rowset
&FLIGHTPLAN_RETURN = &return_mesages [2].GetRowset();
&temp = &return_mesages [2].GenXMLString();
```

Related Links

“SyncRequest” (PeopleCode API Reference)

Sending and Receiving Large Segmented Messages Using Parallel Processing

This section discusses how to:

- Use the OnPreNotify and OnPostNotify PeopleCode events.
- Using the bulk load handler to process large message segments in parallel.
- Select the unordered segments option on the Routings-Routings Definition page.
- Assign service operations to long-running event queues.

Understanding Sending and Receiving Large Segmented Messages Using Parallel Processing

Using parallel processing to send and receive asynchronous service operations that contain large message segments increases integration through-put time.

In sequential processing service operations are processed in order, where the processing of one message must complete before the next message in the queue is processed. In parallel processing, messages are processed in unordered queues, as resources are available for processing.

You can improve system performance and processing times by performing any of these actions.

- Use the OnPreNotify and OnPostNotify PeopleCode events.
- Use the Bulk Load Handler.
- Select the Unordered Segments option on the Routings–Routing Definition page.
- Load balance these types of service operations using long-running event queues.

The only option you must set to send and receive segmented messages using parallel processing is the Unordered Segments option on the Routings–Routing Definitions page. The other options discussed in this section enhance the processing of these types of service operations.

Using the OnPreNotify and OnPostNotify PeopleCode Events

The OnPreNotify and OnPostNotify service operation handler events enable you to perform pre- and post-processing actions on a service operation.

Use the OnPreNotify event to perform high-level pre-processing on data. Typically, this event is used to truncate the database table(s) if a destructive load, versus an update, is required. Note that you cannot use this event to modify actual message data.

Use the OnPostNotify event to perform actions such as start an application engine program, send an email message to indicate the transaction has been completed, and so on.

When implemented, the Broker handler first processes the OnPreNotify PeopleCode event. Then, the system creates one subscription contract for each message segment. The system processes the

subscription contracts in parallel, except for the OnPostNotify contract. The OnPostNotify contract runs only after all subscription contract(s) are completed .

Using the Bulk Load Handler to Process Large Message Segments in Parallel

Using the bulk load handler is a good option when you are processing large messages and large message segments.

The Bulk Load Handler page (IB_SERVICEHDL5_SEC) used to configure a bulk load handler has a field called Truncate Table(s) that provides the option to truncate or not truncate tables. When using the OnPreNotify or OnPostNotify events, clear the Truncate Table(s) option so that tables are not truncated.

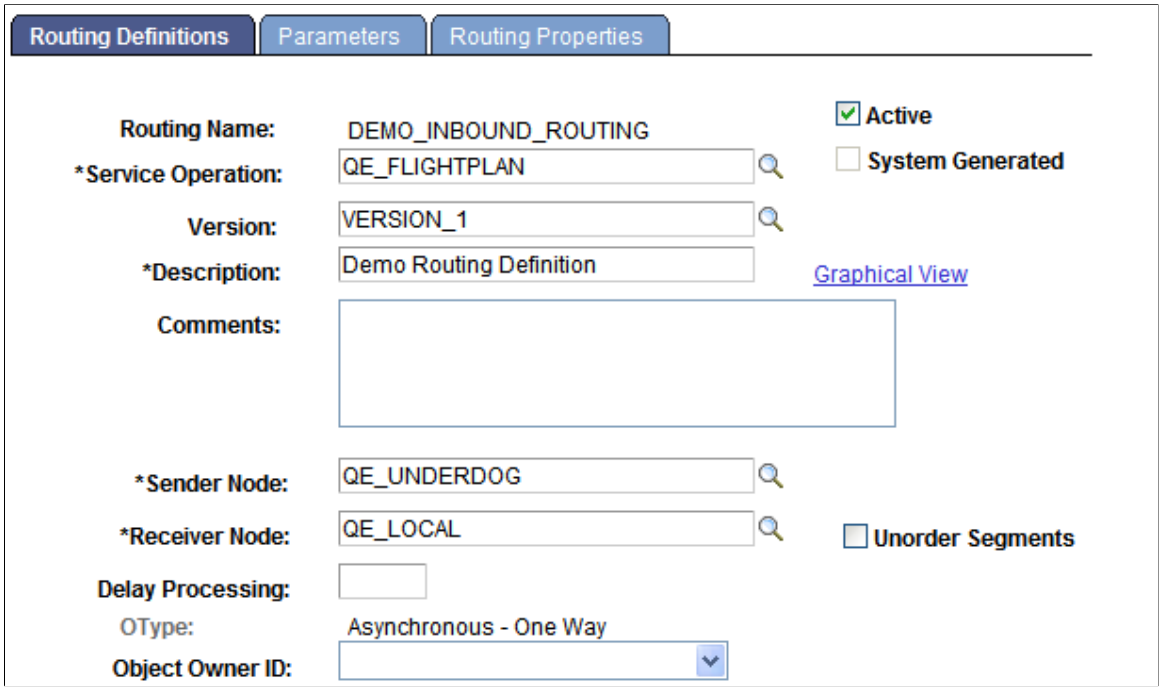
Related Links

"Enabling Table Truncation" (Integration Broker)

Selecting the Unordered Segments Option on the Routings-Routings Definition Page

The Routings – Routings Definition page (IB_ROUTINGDEFN) features an **Unordered Segments** option. The option appears only when working with an inbound routing definition for an asynchronous service operation. To access the page select **PeopleTools** > **Integration Broker** > **Integration Setup** > **Routing Definitions.**

This example illustrates the Routings – Routing Definitions page.



The **Unordered Segments** option appears to the right of the **Receiver Node** field. When you select the **Unordered Segments** option for an inbound routing definition, the system creates a subscription contract for each segment and processes the contracts in parallel.

Assigning Service Operation that Contain Large Segmented Messages to Long-Running Event Queues

For better performance and through-put times, you can load balance the processing of service operations that contain large segmented messages by assigning them to long-running event queues.

Setting up primary-secondary load balancing for long-running events is described elsewhere in the product documentation.

See Implementing Primary-Secondary Dispatchers

Implementing Exception Handling for Synchronous Message Processing

When a an outbound synchronous request fails you can throw a framework exception leading to a message box error and subsequent component roll back of the transaction.

Note: This type of exception handling applies to outbound synchronous requests only, including outbound multi-threaded synchronous requests.

For example, if *10* synchronous requests are performed in parallel (threaded sync request), you have the option to select the **User Exception** check box on the routing definition for the service operation. When the **User Exception** check box is selected, if any of the synchronous requests error, the component is not rolled back. You can check each synchronous request to determine if there is an error and actually read the associated error message. You can then throw an exception or go on to process the next synchronous request in the array.

See “Understanding Routing Definitions” (Integration Broker).

The following example shows sample PeopleCode to read the exception:

```
Local Rowset &FLIGHTPLAN, &FLIGHTPLAN_RETURN;
Local array of Message &messages;
Local array of Message &return_mesages;

&messages = CreateArrayRept(&MSG, 2);
&return_mesages = CreateArrayRept(&MSG, 2);

QE_FLIGHTDATA.QE_ACNUMBER.Value = QE_FLIGHTDATA.QE_ACNUMBER + 1;

&FLIGHT_PROFILE = GetLevel0();

&rs1 = &FLIGHT_PROFILE.GetRow(1).GetRowset(Scroll.QE_NAVIGATION);
&rs2 = &FLIGHT_PROFILE.GetRow(1).GetRowset(Scroll.QE_RADAR_PRESET);
&rs3 = &FLIGHT_PROFILE.GetRow(1).GetRowset(Scroll.QE_ARMAMENT);
&messages [1] = CreateMessage(Operation.SYNC_PARTS);

For &i = 1 To &messages [1].PartCount

   If &i = 1 Then
      &rs1.CopyTO(&messages [1].GetPartRowset(&i));
   End-If;

   If &i = 2 Then
      &rs2.CopyTO(&messages [1].GetPartRowset(&i));
   End-If;
```



```
   If &i = 3 Then
      &rs3.CopyTO(&messages [1].GetPartRowset(&i));
   End-If;

End-For;

&messages [2] = CreateMessage(Operation.SYNC_PARTS);

For &i = 1 To &messages [2].PartCount

   If &i = 1 Then
      &rs1.CopyTO(&messages [2].GetPartRowset(&i));
   End-If;

   If &i = 2 Then
      &rs3.CopyTO(&messages [2].GetPartRowset(&i));
   End-If;

   If &i = 3 Then
      &rs2.CopyTO(&messages [2].GetPartRowset(&i));
   End-If;

End-For;

&return_mesages = %IntBroker.SyncRequest(&messages);

If &return_mesages [1].ResponseStatus = %IB_Status_Success Then

   For &i = 1 To &return_mesages [1].PartCount

//perform local processing on response data

   End-For;

Else

   &nMsgNumber = &return_mesages [1].IBException.MessageNumber;
   &nMsgSetNumber = &return_mesages [1].IBException.MessageSetNumber;
  &exceptString = &return_mesages [1].IBException.ToString();
// Evaluate exception and throw error if necessary

End-If;

If &return_mesages [2].ResponseStatus = %IB_Status_Success Then

   For &i = 1 To &return_mesages [2].PartCount

  //perform local processing on response data   End-For;

  Else

   &nMsgNumber = &return_mesages [2].IBException.MessageNumber;
   &nMsgSetNumber &return_mesages [2].IBException.MessageSetNumber;
  &exceptString = &return_mesages [2].IBException.ToString();

// Evaluate exception and throw error if necessary

End-If;
```

Implementing Primary-Secondary Dispatchers

This section provides an overview of primary-secondary dispatching and describes how to:

- Configure dynamic secondary dispatchers.
- Configure static secondary dispatchers.

- Create template secondary domains.
- Implement primary-secondary load balancing.
- Implement deferred primary domain processing.

Understanding Implementing Primary-Secondary Dispatchers

Primary-Secondary dispatching is where a master domain allocates messages to one or more secondary dispatchers for processing. This section provides an overview of primary-secondary dispatcher processing.

Primary-Secondary Dispatcher Processing

A secondary dispatcher processes service operations assigned to it by a primary dispatcher.

A primary domain allocates service operations to a secondary for processing when:

- The primary detects that a secondary dispatcher is active and not busy processing service operations.
- The secondary has an active queue on which the primary is currently processing service operations.

The dispatcher(s) processing in secondary mode then process the allocated service operations.

Primary and secondary dispatchers can reside on the same or on different machines.

Note: When primary-secondary processing is implemented, there can be only one primary domain at a given time.

You can create a domain consisting of only dedicated secondary pub/sub servers. These servers register themselves as secondaries, along with additional configurable information, such as the number of process handlers booted, so that the appropriate primary server can use that information to allocate work (service operations to process) to the secondary server(s).

The primary domain can allocate work to one or more secondary domains.

Secondary Types

There are two types of dispatcher secondaries:

Term	***Definition***
Dynamic secondaries	A dynamic secondary can change from a primary to a secondary. Dynamic secondaries are configured in conjunction with domain failover. If a secondary domain has the highest priority within a failover group, it can dynamically change to a primary during failover. You configure dynamic secondaries in the Failover Configuration page in the Service Operations Monitor.

Term	*Definition*
Static secondaries	Static secondaries are those that cannot become primaries without manual configuration. You configure static secondary domains in PSADMIN.
Template secondaries	A template secondary is an already-configured primary domain that you import into PSADMIN an save as a secondary domain. Template secondaries enable you to dynamically add secondary domains without performing any configuration changes in PSADMIN. You use the Import Domain Configuration command in PSADMIN to import a primary domain configuration and then save it as a static secondary domain. The secondary domain created uses all the Pub/Sub processes and queue lists configured for the primary domain on which the secondary template is based. If dedicated servers are configured for the primary domain, they are also imported an available on the secondary domain.

Failover and Primary-Secondary Dispatchers

You can create a secondary domain for use in domain failover.

The domain with the highest priority dynamically becomes the active domain (primary domain) in each group during failover. The next domain in priority will be programmatically configured as an active secondary domain.

When a failover occurs the domain that failed becomes inactive. The failover domain specified goes from an active secondary to an active primary. The next domain in priority then becomes an active secondary.

You can set failover for secondary dispatchers. However, secondary dispatchers cannot be part of any group and you cannot prioritize them.

See Setting Up Domain Failover.

Configuring Dynamic Secondary Dispatchers

Use the Failover Configuration page in the Service Operations Monitor to configure dynamic secondary dispatchers.

See Setting Up Domain Failover.

Creating Template Secondary Domains

This section discusses how to:

- Import domain configurations from application domains.
- Import domain configurations from files.
- View template secondary domains.
- Add and delete dispatcher queues from template secondary domains.
- Restore dispatcher queue lists.

Understanding Template Secondary Domains

Template secondary enable you to dynamically add secondary domains without performing any configuration changes in PSADMIN.

When you create a template secondary domain you import a domain configuration and save it as a template for the secondary domain. This process creates a static secondary domain that uses all of the pub/sub processes and queue lists configured for the domain that you import. If dedicated servers are configured for the domain that you import, they are imported and available on the secondary domain. After you import a domain configuration and save it as a secondary domain, you can add or remove dispatcher queues from the secondary domain as needed.

All template secondary domains must be based on the same primary domain configuration.

Template Secondary Domain Types

There are two options for importing a domain as a template:

Field or Control	***Description***
IB Secondary Basic	When the configuration is imported, all the Pub/Sub processes are configured identically to the domain on which the secondary template is based, including the PSWATCHSRV and PSMONITOR server processes. Other process, such as PSAPPSRV, PSSAMSRV, and so on, are not included in the template secondary domain. However, you can modify the template secondary domain configuration file to include these processes if needed
IB Sync Secondary	As in the IB Secondary Basic template, when the configuration is imported, all the Pub/Sub processes are configured identically to the domain on which the secondary template is based, including the PSWATCHSRV and PSMONITOR server processes. However, unlike the IB Secondary Basic template, this option imports the PSAPPSRV, JSL, and JREPSVR processes from the domain on which the template is based. In addition you have the option to change the default Jolt port (Jolt port taken from the primary configuration file).

Understanding Importing Domain Configurations

To import the domain configuration on which the template secondary domain is based, you use the Import Domain Configuration command in PSADMIN. You can import a domain that is already configured in PSADMIN or you can import a domain configuration from a file.

To import a domain configuration from an application domain you specify the location of <PS_CFG_HOME> for the domain that you want to import. For example, the location might be *c:\documents and settings\admin\psft\ps\<PS_CFG_HOME>*.

To import a PeopleTools 8.49 or earlier application domain you must specify the <PS_HOME> location for <PS_CFG_HOME>. For example, the location might be *c:\documents and settings\<PS_HOME>\<PS_CFG_HOME>*.

Prerequisites for Importing Domain Configurations

If you are importing a domain that is already configured in PSADMIN, you must first set the PS_FILEDIR environment variable equal to the PS_HOME location of the domain you importing.

If importing a domain configuration from a file, you must first set the PS_FILEDIR environment variable to the location where you are importing the file.

See “Understanding Setting PS_FILEDIR, PS_SERVDIR, and PS_TREEBASEDIR Environment Variables” (Integration Broker).

Importing Domain Configurations from Files

To import a domain configuration from a file:

1. Open PSADMIN.

   The PeopleSoft Server Administration menu appears.

2. Enter *1* for Application Server and press the **ENTER** key.

   The PeopleSoft Application Server Administration menu appears.

3. Enter *4* for Import Domain Configuration and press the **ENTER** key.

   The PeopleSoft Import Application Server Configuration menu appears.

4. Enter *2* for Import IB Primary Configuration and press the **ENTER** key.

   A Configuration Templates prompt appears.

5. Select one of the following options:

   - Enter *1* to import the domain as an IB Secondary Basic template and press the **ENTER** key.
   - Enter *2* to import the domain as an IB Sync Secondary template and press the **ENTER** key.

   The PeopleSoft Import Application Server Configuration menu appears.

6. Enter *1* for Import from file and press the**ENTER** key.

   A prompt displays to enter the full path to the domain configuration file to import.

7. Enter the full path to the domain configuration file to import and press the **ENTER** key.

   A prompt displays to enter a name for the new domain.

8. Enter a name for the new domain and press the **ENTER** key.

The system merges the domain configuration with the new template secondary domain and creates the new configuration and loads it on the application server. Upon completion, the PeopleSoft Domain Administration menu appears, where you may boot the template secondary domain, configure the template secondary domain or perform other administrative tasks.

Importing Domain Configurations from Application Domains

To import a domain configuration from an application domain:

1. Open PSADMIN.

   The PeopleSoft Server Administration menu appears.

2. Enter *1* for Application Server and press the **ENTER** key.

   The PeopleSoft Application Server Administration menu appears.

3. Enter *4* for Import Domain Configuration and press the **ENTER** key.

   The PeopleSoft Import Application Server Configuration menu appears.

4. Enter *2* for Import IB Primary Configuration and press the **ENTER** key.

   A Configuration Templates prompt appears.

5. Select one of the following options:

   - Enter *1* to import the domain as an IB Secondary Basic template and press the **ENTER** key.
   - Enter *2* to import the domain as an IB Sync Secondary template and press the **ENTER** key.

   The PeopleSoft Import Application Server Configuration menu appears.

6. Enter *2* to for Import from application domain and press the**ENTER** key.

   A prompt displays to enter the location of <PS_CFG_HOME>.

7. Enter the location of <PS_CFG_HOME> and press the**ENTER** key.

   The Tuxedo Domain List appears that lists the application domains that you can import.

8. Enter the number that corresponds to the application domain to import.

   A prompt displays to enter a name for the new domain.

9. Enter a name for the new domain and press the **ENTER** key.

The system merges the domain configuration with the new template secondary domain and creates the new configuration and loads it on the application server. Upon completion, the PeopleSoft Domain Administration menu appears, where you may boot the template secondary domain, configure the template secondary domain or perform other administrative tasks.

Adding and Removing Dispatcher Queues from Template Secondary Domains

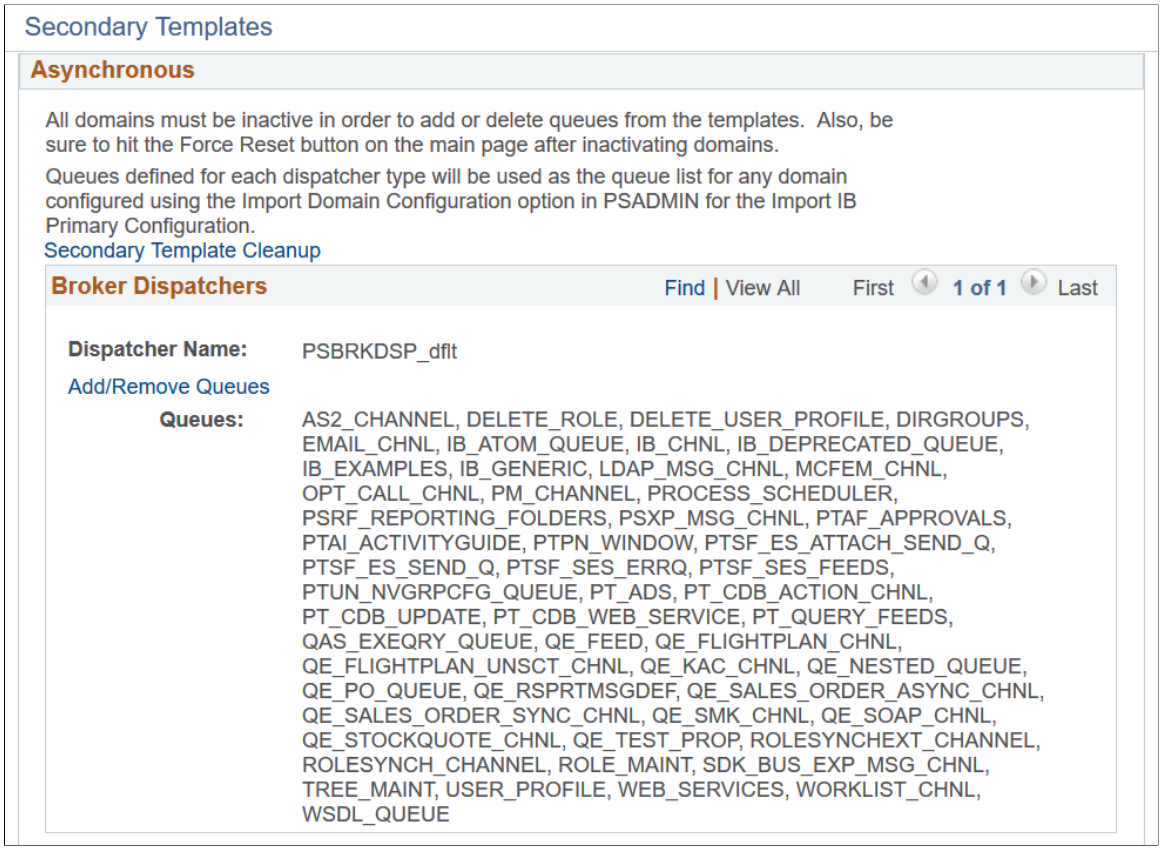
This section applies only to asynchronous secondary templates.

Template secondary domains contain all of the dispatcher queues that exist on the domain on which it is based. However, you can add and remove queues to configure the template secondary domain to suit your requirements.

Note: Before you can add or remove queues from a template secondary, you must inactivate all domains.

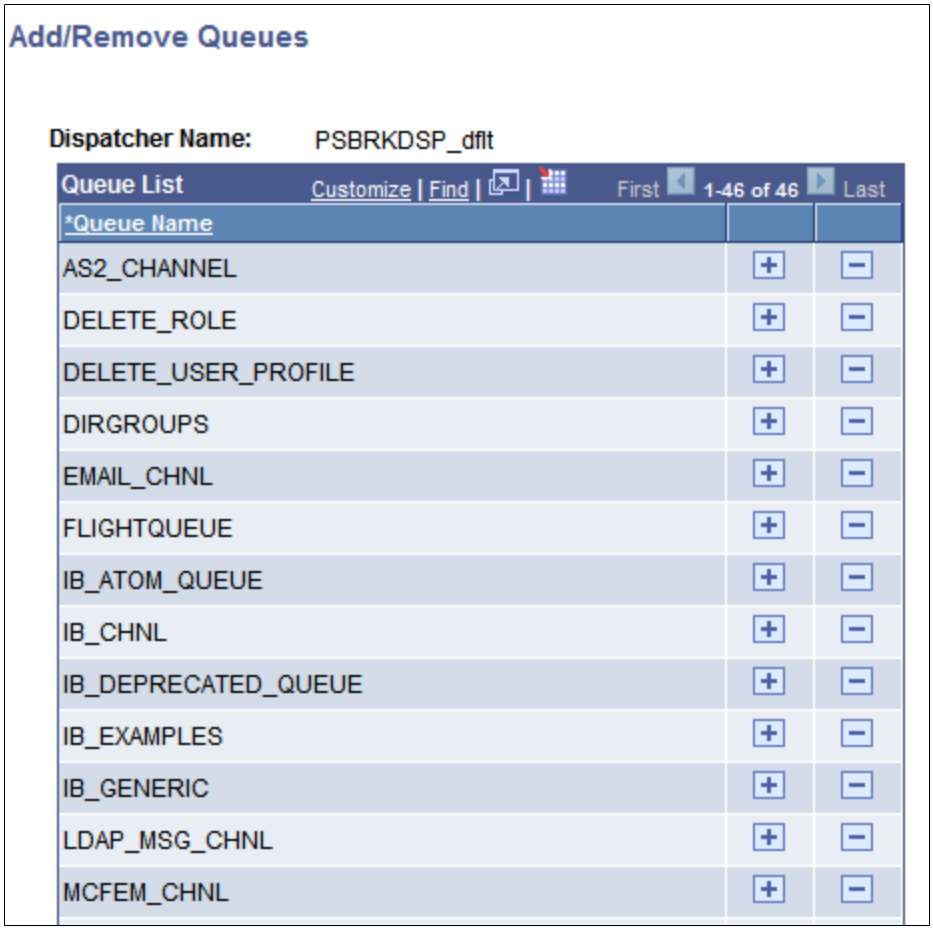
The Secondary Templates page (IB_DOMAIN2_SEC) lists the dispatcher queues assigned to each dispatcher process of the template secondary.

This example illustrates the Secondary Templates page. Use this page to view the queues assigned to each dispatcher process of a template secondary.



From the Secondary Templates page, you can use the **Add/Remove Queues** link located under each dispatcher process name to access the Add/Remove Queues page to add or remove queues assigned to the dispatcher.

This example illustrates the Add/Remove Queues page. Use this page to add or remove queues for a dispatcher process of a template secondary.



Note that this example shows a partial queue list for the domain.

On the Add/Remove Queues page you can use the plus (+) button to add a queue to the queue list. Use the minus (-) button to remove a queue from the list.

To add or remove dispatcher queues for template secondary:

1. Access the Domain Status page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status**).
2. Check the **All Domains Inactive** box and click the **Update** button.

   The **Force Reset** button appears.
3. Click the **Force Reset** button to reset any contacts that are in a *Started* or *Working* state.
4. Click the **Secondary Templates** link.

   The Secondary Templates page appears.
5. Click the **Add/Remove Queues** link for a queue dispatcher.

   The Add/Remove Queues page appears and displays the queue list for the dispatcher.

6. To add a queue:
   a. Click the plus (+) button to insert a new row into the list.
   b. Click the **Lookup** button to search for a queue to add.
   c. Click the **OK** button at the bottom of the Add/Remove Queues page.

      The Secondary Templates page appears.
   d. Click the **Update** button.

      The Domain Status page appears.

7. To delete a queue:
   a. Click the minus (-) button next to the queue to delete.
   b. Click the **OK** button in the dialog box to confirm the delete action.
   c. Click the **OK** button at the bottom of the Add/Remove Queues page.

      The Secondary Templates page appears.
   d. Click the **Update** button.

      The Domain Status page appears.

8. Activate the domains in the messaging system.

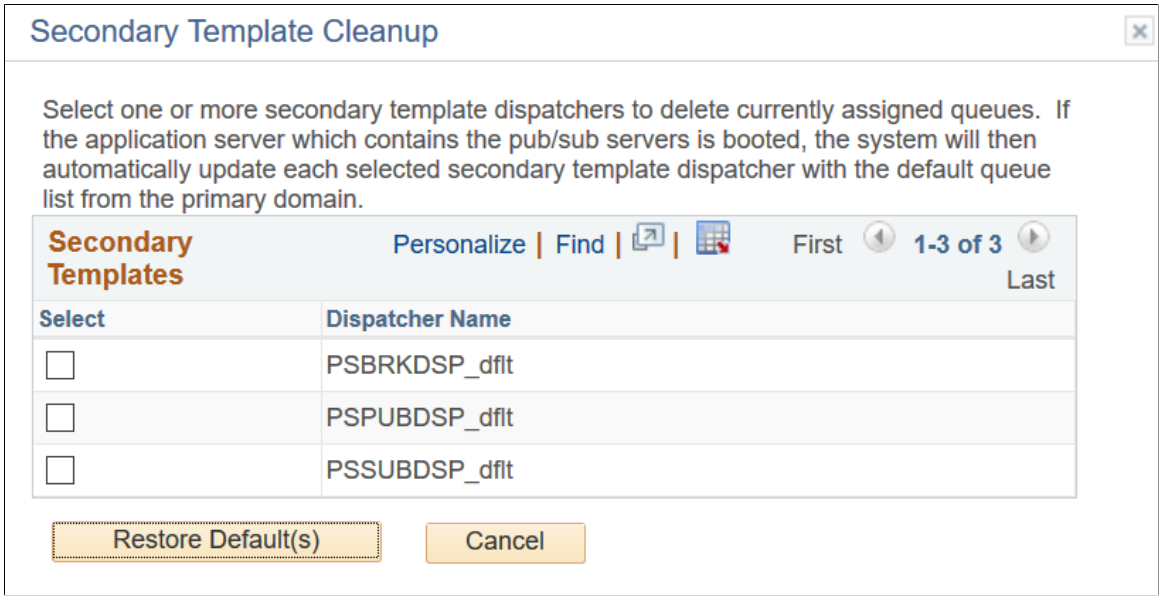
   Check the **All Domains Active** box and click the **Update** button to activate the domains.

Restoring Template Secondary Dispatcher Queue Lists

Restoring a queue list deletes any changes you have made to a template secondary dispatcher queue list, and restores it to the queue list that you originally imported from the primary domain.

To restore a template secondary queue list, use the Secondary Template Cleanup page (IB_DOMAIN3_SEC).

This example illustrates the Secondary Template Cleanup page. Use this page to restore dispatcher queue lists to the default settings from the primary domain.



In the previous example, PSBRKDSP_dflt appears in the **Dispatcher Name** field, meaning that additions or deletions have been made to the queue. In this example, you could use the page to restore the default template secondary dispatcher list for PSBRKDSP_dflt.

To restore a template secondary dispatcher queue list, you must first inactivate all domains on the system.

To restore a template secondary dispatcher queue list:

1. Inactivate the domains on the messaging system:
    a. Access the Domain Status page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status**).
    b. Check the **All Domains Inactive** box and click the **Update** button.

       The **Force Reset** button appears.
    c. Click the **Force Reset** button to reset any contacts that are in a *Started* or *Working* state.

2. Access the Secondary Template Cleanup page:
    a. From the Domain Status page, click the **Secondary Templates** link.

       The Secondary Templates page appears.
    b. On the Secondary Templates page, click the **Secondary Template Cleanup** link.

       The Secondary Template Cleanup page appears.

3. Check the box next to each dispatcher process name for which you want to restore the default queue list.
4. Click the **Restore Default(s)** button.

The Domain Status page appears.

5. Activate the domains in the messaging system.

   Check the **All Domains Active** box and click the **Update** button to activate the domains.

Implementing Primary-Secondary Load Balancing

This section provides and overview of primary-secondary load balancing and discusses how to set up primary-secondary load balancing on the PeopleSoft system.

Understanding Primary-Secondary Load Balancing

You can implement primary-secondary load balancing on the integration system to compensate for processing capabilities of various machines on which primary domains and secondary domains run.

As an example, you might have a domain on machine that is also running the PeopleSoft Pure Internet Architecture. In this case, you could configure primary-secondary load balancing such that the machine that is running the PeopleSoft Pure Internet Architecture processes fewer requests than other machines on which domains reside.

Another example is a situation where the machines on which you are running domains have different processing capabilities due to the hardware installed in them. In this situation you can configure the machines with the most process power to process the greater number of requests.

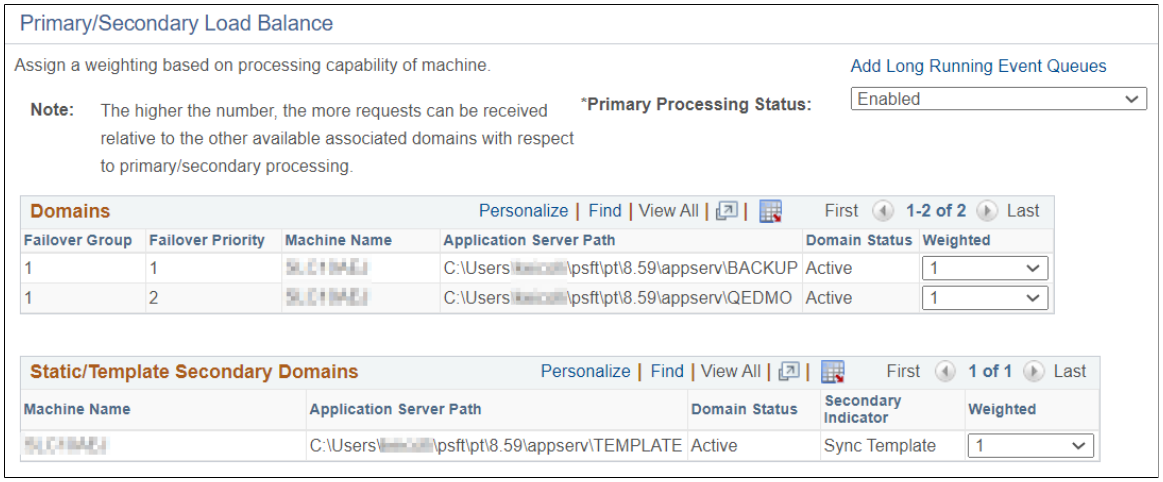
To configure primary-secondary load balancing, you assign a weight between *1* and *10* to each domain to distribute request processing. A domain assigned a weighted value of *1* processes the fewest requests; a domain assigned a weighted value of *10* processes the greatest number of requests.

Setting Up Primary-Secondary Load Balancing

You set up primary-secondary load balancing using the Primary/Secondary Load Balancing page (IB_DOMAIN_SEC).

To set up primary-secondary load balancing, you assign a processing weight value to each domain. The domain with the lowest number processes the fewest number of requests and is the primary domain. The domains with the higher numbers are the secondary domains and process the greatest number of requests.

This example illustrates the Primary/Secondary Load Balance page. The example shows primary-secondary load balancing set up for the system.



The Domain section on the page lists information for the primary domain, while the Static/Template Secondary Domains section lists information about static secondary domains.

The example shows two domains configured on one machine. The domains listed in the Domains section, *BACKUP* and *QEDMO*, are the primary domains and have a load balance weight of *1* assigned to it. Given the load balance weight assigned to the domains, they processes the fewest number of requests.

The domain listed in the Static/Template Secondary Domains section, *TEMPLATE*, has a load balance weight of *1* assigned to it. It processes the same number of requests as the master domain.

To set up primary-secondary load balancing:

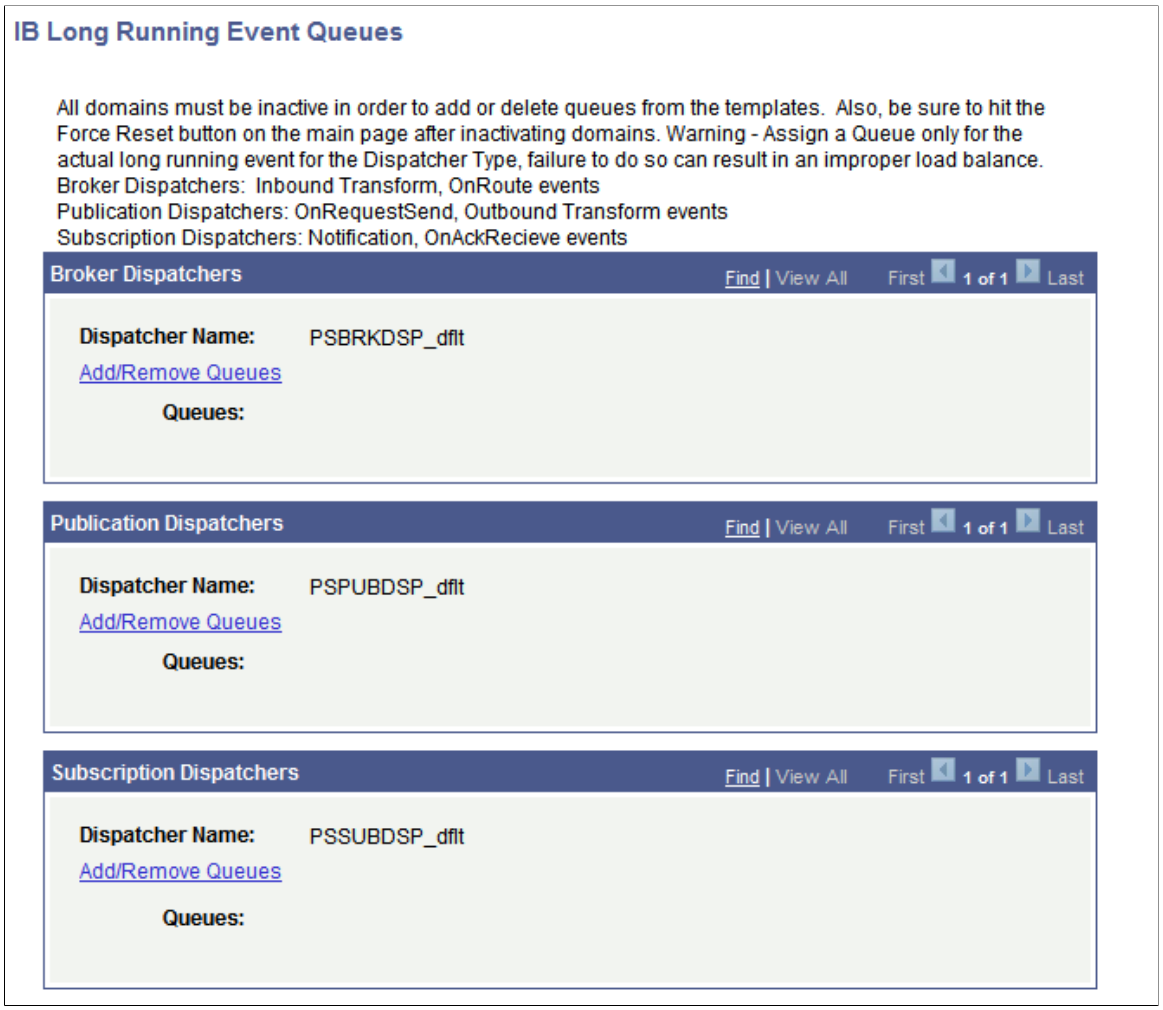
1. Access the Primary/Secondary Load Balance page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status** and click the **Primary/Secondary Load Balance** link.).
2. For each domain select a value from the **Weighted** drop-down list box to assign a load balancing weight for the domain.
3. Click the **OK** button.

Setting Up Primary-Secondary Load Balancing for Long-Running Events

Typical primary-secondary load balancing usually focuses on normal events running at a high throughput. However, there are some instances where there are low volume service operation that contain long-running events. For better performance, you can assign these types of service operation to long-running event queues, where the processing is spread across all potential master and Secondary handlers.

You use the IB Long Running Event Queues page (IB_DOMAIN4_SEC) to assign service operations to long-running event queues.

This example illustrates the IB Long Running Event Queues page.

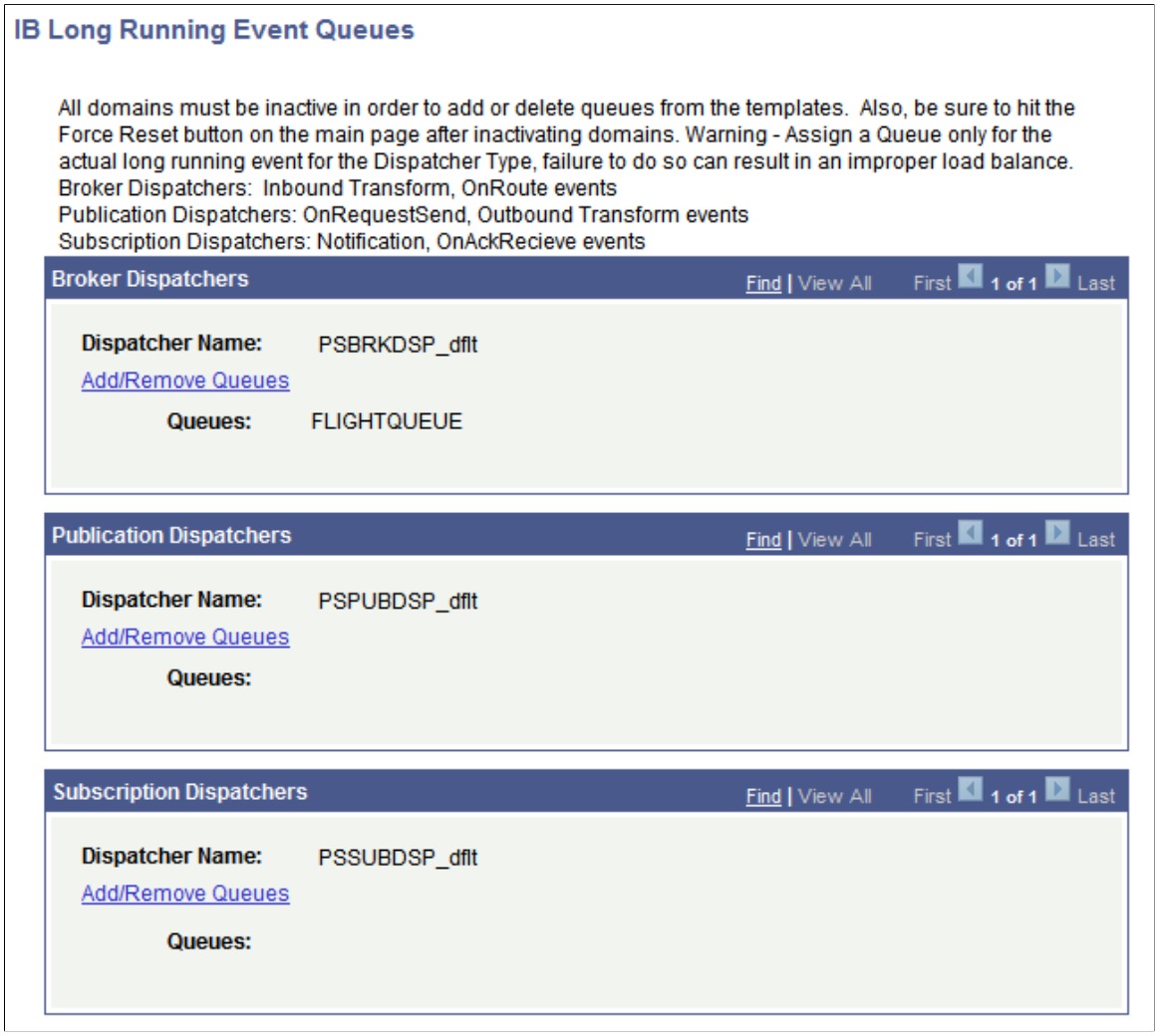


To assign a service operation to a long-running event queue, you add the service operation queue that is currently defined on the service operation definition to one of the dispatcher queues on the page. When you click the **Add/Remove Queues** link for one of the dispatchers, the Add/Remove Queues page (PSIBQUEUE_SEC) appears.

This example illustrates the Add/Remove Queue page.

You use the Lookup button on the page to search for the service operation queue to which the service operation is assigned. When you click the OK button, the IB Long Running Event Queues page appears and the queue you specified appears in the dispatcher **Queue** field.

This example illustrates the IB Long Running Event Queues page. The example shows the service operation queue *FLIGHTQUEUE* added to the Broker Dispatcher queue.



The following table describes the proper dispatchers to which to assign service operation queues based on the type of processing required:

Dispatcher	***Processing***
Broker Dispatcher	• Inbound transforms • OnRoute events
Publication Dispatcher	• Notifications • OnAckRecieve events
Subscription Dispatcher	• Outbound transforms • OnSend events

Important! Do not assign service operations that contain long-running events to the same queues as those that do not contain long-running events. Processing performance for the normal high-volume service operations can be impacted.

To assign a service operation to a long-running event queue:

1. Access the Domain Status page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status**).

   The Domain Status page appears.

2. Inactivate the domain.

   In the Domains grid, locate the **Domain Status** drop-down list box and select Inactive.

3. Click the **Force Reset** button.

4. Click the **Primary/Secondary Load Balance** link.

   The Primary/Secondary Load Balance page appears.

5. Click the **Add Long Running Event Queues** link.

   The IB Long Running Event Queues page appears.

6. Click the **Add/Delete Queue** link for the dispatcher to which to add the service operation queue.

   The Add/Remove Queues page appears.

7. Enter a name or use the **Lookup** button to search for the name of the queue to which the service operation is assigned.

8. Click the **Update** button.

   The Primary/Secondary Load Balance page appears.

9. Click the **OK** button.

   The Domain Status page appears.

10. Activate the domain by selecting Active from the **Domain Status** drop-down list in the Domains grid.

11. Click the **Update** button.

Implementing Deferred Primary Domain Processing

This section provides an overview of deferred master domain processing and discusses how to set up deferred primary domain processing.

Understanding Deferred Primary Processing

PeopleSoft Integration Broker enables you to defer request processing on primary domain to secondary domains that are available for processing. Configuring deferred primary processing enables you to free processing resources on the primary domain machine due to hardware or processing power limitations, or so it can run other processes.

The Primary/Secondary Load Balance page features a **Primary Processing Status** drop-down list box where you set the processing status for the primary domain. The following table lists the primary processing statuses and their descriptions.

Master Domain Processing Status	*Description*
Enabled	The primary domain processes its appropriate share of requests. (Default.)
Deferred – All Queues	The primary domain does not send any requests to its respective process handler(s) as long as there is at least one active secondary domain that can be used for the dispatch cycle.
Deferred – Unordered Queues	The primary domain does not send any requests in an unordered queue to its respective process handler(s) as long as there is at least one active secondary domain that can be used for the dispatch cycle. When you select this option the primary domain dispatchers only send requests to a secondary domain for processing if the queue being processed is defined as unordered queue. If the queue is not unordered, the primary domain sends the request to its own process handler for processing not to the secondary domain.

If the system is set to any of the deferred modes the primary will process requests if no secondary dispatchers are available. In each of the deferred modes, the primary assigns processing to secondary dispatchers based on the load balancing weight value assigned to the secondary dispatcher.

Setting Up Deferred Primary Domain Processing

To set up deferred primary domain processing:

1. Select **PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status**.

   The Domain Status page appears.

2. Click the **Primary/Secondary Load Balance** link.

   The Primary/Secondary Load Balance page appears.

3. From the **Primary Processing Status** drop-down list box, select a primary domain processing status.

   The valid values are:

   - *Enabled*
   - *Deferred — All Queues*

- *Deferred — Unordered Queues*

4. Click the **OK** button.

Allowing Multiple Active Domains

By default, only one domain may be active in the Integration Broker system. However, PeopleSoft provides the option to enable the activation of multiple domains.

To allow multiple active domains, use the Monitor Setup Options page in the Service Operations Monitor.

To allow multiple active domains:

1. Access the Monitor Setup Options page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Monitor Setup Options**).
2. Select the **Allow Multiple Active Domains** option.
3. Click the **Save** button.

Setting Up Domain Failover

This section discusses how to:

- Enable failover on domains.
- Set up dynamic primary—secondary dispatchers.
- Check the validity of queue sets.
- View queues assigned to failover groups.
- Schedule pause times for failover.

Understanding Domain Failover

This section discusses domain failover.

Domain Failover

Domain failover ensures that PeopleSoft Integration Broker continues processing message requests and responses, even if it incurs errors or other problems on the primary domain. When failover is activated, service operation processing will switch to back up domains should Integration Broker incur any errors or problems on the primary domain. In addition, should the domain fail, you can send a system-generated email notification to individuals.

If the connection with the database is lost and the handlers are processing service operations at that time, the handlers attempt to reboot. If initialization fails, the handlers are not rebooted. If you are using failover, the failover process takes over and failover of the domain occurs.

Domain Failover Groups

You can set up domain failover groups, so that all failover takes place on specific domains. To set up failover groups, you assign a domain a failover group number. After you assign domains to a group, you then assign the failover priority for all domains within the group.

Note: If you do not use dedicated messaging servers, you typically do not need to use failover groups.

Note: Queue sets within failover groups must be identical; queue sets between failover groups must be unique.

The example of the Failover Configuration page in the Setting Up Dynamic Primary-Secondary Dispatchers section of the documentation shows four domains attached to the application server. The first two domains have been assigned to failover group one, as indicated by the value *1* in the **Failover Group** field for each domain. The last two domains have been assigned to failover group two, as indicated by the value *2* in the Failover Group field for each domain.

The failover priority within group one is as follows: the first domain in the list is the main and primary back-up domain as indicated by the failover priority value *1*; the second domain in the list is the second back-up domain as indicated by the failover priority value *2*.

The failover priority within group two is as follows: the third domain in the list is the first back-up domain for group two as indicated by the failover priority value *1*; the last domain in the list is the second back-up domain as indicated by the failover priority value *2*.

Failover Groups — Priority Reset

PeopleSoft Integration Broker features a priority reset option that works in conjunction with failover groups.

When you set this option and failover occurs, the system attempts to use the domain defined as group priority value of 1 before failing over to the next sequential domain.

As an example, you could have a failover group defined as follows:

Group Failover Priority	***Domain Name***
1	QEDMO
2	DOMAIN02
3	DOMAIN03

In the example in the table, domain *QEDMO* has already failed over to domain *DOMAIN02*. If domain *DOMAIN02* subsequently fails, the system attempts to failover back to the domain with the highest failover priority setting, domain *QEDMO*. If the system is unsuccessful in failing over to domain *QEDMO*, domain *DOMAIN02* fails over to the next sequential domain in the failover group, domain *DOMAIN03*.

You set the **Priority Reset** option on the Failover Configuration page.

See Enabling Failover on Domains.

Dynamic and Static Primary-Secondary Dispatchers

You can implement primary-secondary dispatchers in conjunction with domain failover.

When *dynamic* secondaries are implemented, the domain with the highest priority will become the active domain (primary domain) in each group during failover. The next domain in priority is automatically programmatically configured as an active secondary domain. You configure dynamic secondaries in the Service Operations Monitor.

Static domain secondaries are always secondaries. You configure static secondaries in PSADMIN.

Failover Priority — General Failover

In general domain failover, if a failover domain becomes inactive, the system attempts failover to the next in priority domain. If it is unable to do so, the next domain in priority becomes the active domain.

As an example, consider an integration system with the domains and failover priorities shown in the following table:

Domain	***Failover Priority***
Domain A	1
Domain B	2
Domain C	3

In this integration system if Domain A fails, the system will failover to Domain B. If Domain B later fails, the system will failover to Domain C.

Failover Priority — Dynamic Secondary Failover

In dynamic secondary failover, if the dynamic secondary fails, the system generates and email notification and the dynamic secondary becomes inactive. The system does not failover to find another dynamic secondary.

Failover on the primary system has to occur for a dynamic secondary to automatically go into effect.

Failover Priority Modification

If you modify failover priorities when failover is enabled and change the priorities of the current active domain, all domains are reset to inactive and the domain with the priority value of *1* is activated. However, if failover is not active and you change priorities, PeopleSoft Integration Broker saves the changes without any domain status reset.

Failover and Node Pause Times

Domain failover is disabled during node/system pause times. Additional information is provided elsewhere in the product documentation.

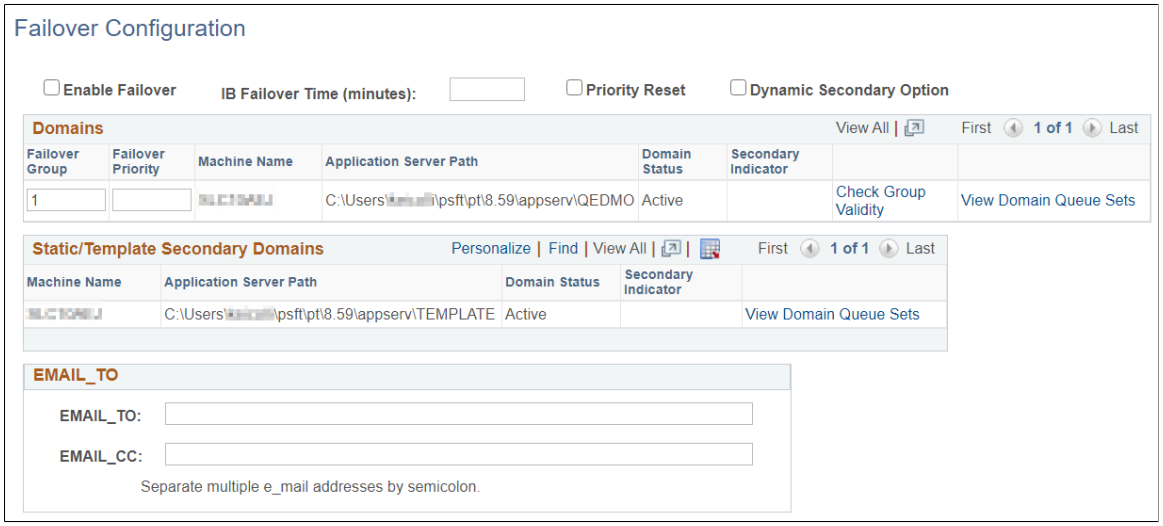
You can add scheduled pause times to nodes during which time failover does not occur. A typical case for scheduling system pause times is for scheduled or routine system maintenance. Scheduling pause times prevents failover from occurring during the defined time range and notifications from being sent when a domain is brought back up.

See “Understanding Pausing Nodes” (Integration Broker Service Operations Monitor).

Enabling Failover on Domains

Use the Failover Configuration page (IB_AMM_FAILOVER) to enable failover on domains.

This example illustrates the Failover Configuration page.



To set up domain failover:

1. Access the Failover Configure page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status** and click the **Set Up Failover** link).

2. Select the **Enable Failover** box.

3. In the **IB Failover Time (minutes)** field, specify the number of minutes that can pass without the domain registering itself before the failover should commence

4. (Optional.) To implement dynamic secondaries, select the **Dynamic Secondary Option**.

5. (Optional.) In the **Failover Group** field, enter a numeric value to specify a group to which a domain belongs.

   A value of *1* indicates that the domain is the first backup domain; a value of *2* indicates that the domain is the second back-up domain if the first backup domain fails; and so on.

6. In the **Failover Priority** field, enter a numeric value to specify the priority for a back up domain in the failover configuration.

A value of *1* indicates that the domain is the first backup domain; a value of *2* indicates that the domain is the second back up domain; and so on.

7. (Optional.) In the **Email_TO** field, specify the email addresses of people to whom the system sends a notification about the domain failure if it occurs.

   Separate multiple email addresses with a semicolon.

8. (Optional.) In the **Email_CC** field, specify the email addresses of people who receive copies of the domain failure notification.

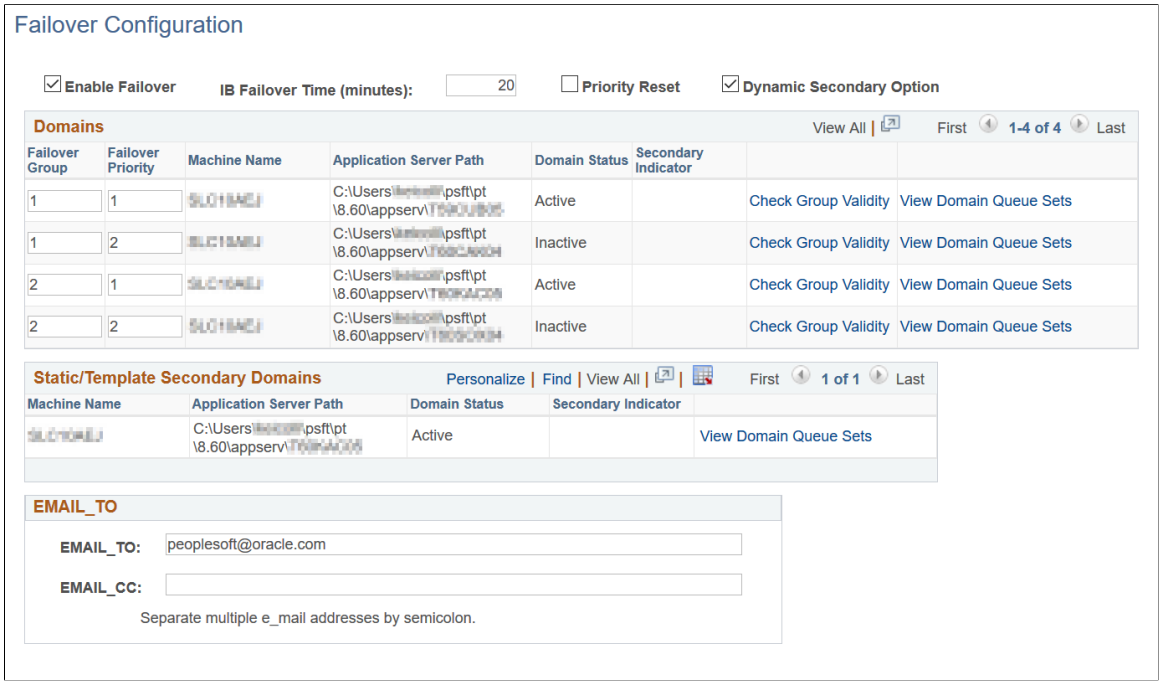
   Separate multiple email addresses with a semicolon.

9. Click the **Save** button.

Setting Up Dynamic Primary-Secondary Dispatchers

When dynamic secondaries have been set the Secondary Indicator column on the Failover Configuration page displays the status *Dynamic* to indicate the domains that are serving as dynamic secondaries.

This example illustrates the Failover Configuration page. In the example, a dynamic secondary is configured, as indicated by the term *Dynamic* appearing in the Secondary Indicator column.



As noted earlier, the domain in a failover group with the highest priority becomes the primary and the domain with the second highest priority becomes the secondary.

To set up primary—secondary dispatchers, follow the procedure for setting up failover and verify that you:

- Check the **Enable Failover** box.
- Check the **Dynamic Secondary Option** box.

- Set up at least one failover group that contains at least two domains.
- Set a failover priority for each domain in the failover group.
- Save your settings.

Checking Queue Validity

Use the **Check Group Validity** link on the Failover Configuration page to verify that all queues assigned to the pub/sub processes in a failover group are the same.

When you click the link a message box appears that indicates if the group is valid.

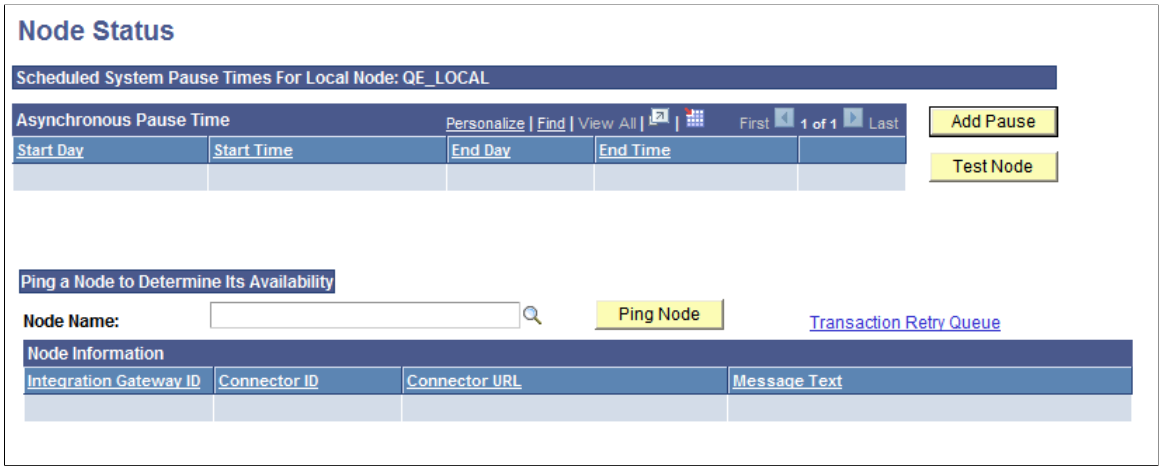
Viewing Queues Assigned to Failover Groups

Use the **View Group Queues** link on the Failover Configuration page to view the queues assigned to each dispatcher in a failover group. Queues must be identical among all groups.

Scheduling Pause Times for Failover

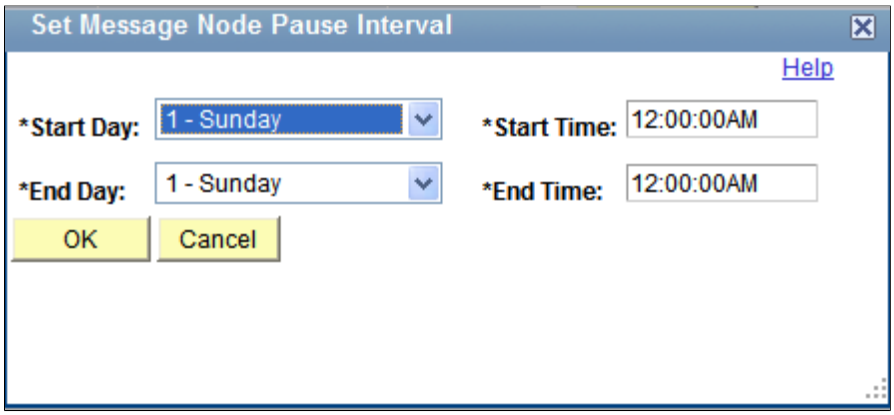
Use the Node Status page (AMM_NODE_STATUS) in the Service Operations Monitor to schedule pause times for failover. To access the page select **PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Node Status.**

This example illustrates the Node Status page. Use this page to schedule pause times for failover.



When you click the **Add Pause** button the Set Message Node Pause Interval page (AMM_ADD_SPTIMES) appears where you can set the system pause start and stop day and time.

This example illustrates the Set Message Pause Time Interval page. Use this page to set pause start and stop times for a node.



To schedule a pause time for failover:

1. Access the Node Status page (**PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Node Status**).
2. Click the **Add Pause** button.

   The Set Message Node Pause Interval page appears.
3. Schedule the pause time to start:

   a. From the **Start Day** drop-down list, select the day of the week for the pause time to start.

   b. In the **Start Time** field, enter the time of day for the pause time to start.
4. Schedule the pause time to end:

   a. From the **End Day** drop-down list, select the day of the week for the pause time to end.

   b. In the **End Time** field, enter the time of day for the pause time to end.
5. Click the **OK** button.

   The Node Status page appears and the details of the scheduled pause time appear in the Asynchronous Pause Time grid.

Configuring Integration Gateways for Load Balancing When Using Third-Party Software

This section discusses how to configure integration gateways in conjunction with using third-party load balancing software.

Understanding Configuring Integration Gateway for Load Balancing When Using Third-Party Software

To increase gateway performance you can use load balancing using third-party software. Load balancing involves the use of a third-party load balancing software product and the installation and configuration of multiple gateways. Then, when messages are sent or published to your messaging system, the load balancing software analyzes the load on installed gateways and determines to which gateway to send the messages to balance the load on all gateways.

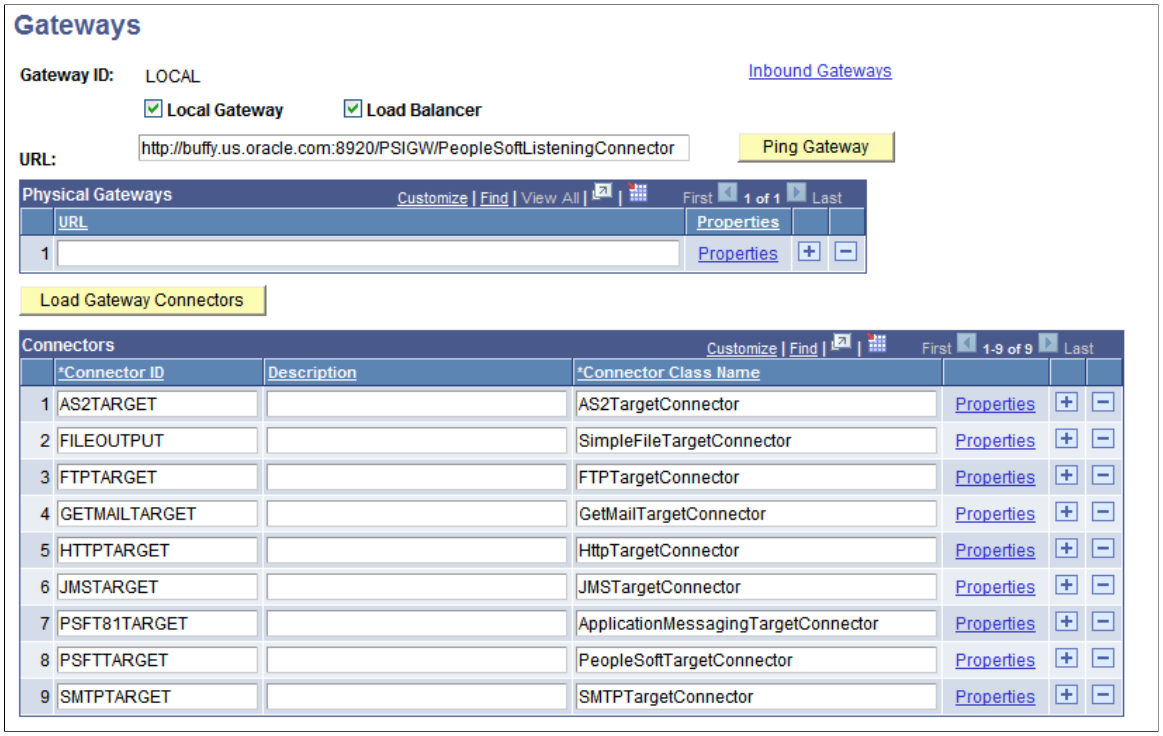
For installation and configuration information about your load balancing software, please see the documentation that is included with the product.

Configuring Load Balancing on Integration Gateways When Using Third-Party Software

To configure gateways participating in load balancing, you must specify the URLs of the gateways in use for load balancing on the Gateways page, and then set integration gateway properties for each gateway you specify. Note that you can set different properties for each gateway.

To access the Gateways page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**. Select the default local gateway.

This example illustrates the Gateways page. In this example, load balancing is enabled, as indicated by the check in the **Load Balancer** box.



To configure an integration gateway for load balancing:

1. Access the Gateways page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways**.

2. Select the default local gateway.
3. Select the **Load Balancer** box.
4. In the Physical Gateway section, in the **URL** field, enter a gateway URL for a gateway that will be used for load balancing.
5. Click the plus (+) button and enter gateway URLs for each additional gateway to be used for load balancing.
6. Click the **Save** button.
7. For each gateway URL entered, click the **Properties** link to set integration gateway properties for that gateway.

Implementing Inbound Request Load Balancing Using Virtual Application Server Domains

This section discusses how to:

- Configure a synchronous secondary template domain.
- Define application server URLs for load balancing.
- Define gateway URLs for inbound processing.
- Define virtual server nodes.
- Register and synchronize integration gateways and virtual application server domains.
- View virtual server domains registered to an integration gateway.
- Enforce secure inbound requests.

Understanding Implementing Load Balancing Using Virtual Application Server Domains

You can configure one or more integration gateways for load balancing using virtual application server domains. This section provides an overview of implementing this process.

Synchronous Secondary Template Domains

To implement load balancing using virtual application server domains you must configure a synchronous secondary template domain in PSADMIN. When the application server boots, the secondary template registers application server information to all gateways defined for inbound processing. You can view the secondary templates that are registered to a gateway using the Inbound Gateways page in the Gateway component.

Note: The terms *secondary template domain* and *secondary template* are used interchangeably in this section.

When the application server is booted, the secondary template domain appears in the Domain Status component in the Service Operations Monitor, where you perform additional configuration as well as manage dispatcher queues.

Load Balancing Application Server URLs

When you boot the secondary template domain, it appears in the Domain Status component in the Service Operations Monitor. Synchronous secondary templates are indicated in the system as *Sync-Template.*

You then use the Secondary Templates page to add the application server URL information (machine name: jolt port) in order for the gateway(s) to make a proper connection to that particular domain.

Gateway URLs and Inbound Gateway Processing

If you are using multiple gateways, you then use the Inbound Gateways page in the Gateways component to add gateway URLs for each physical gateway that you are using. The URL of the default local gateway is populated on the page by default.

You can also use the Inbound Gateways page to view current service URLs assigned to nodes.

Virtual Server Nodes

You can define a virtual server node so that in situations where an inbound request does not specify a "To" node, the system will send the request to a PeopleSoft node defined in the integration gateway properties file.

Secure Inbound Requests

The integration gateway properties file features a property, ig.SecureVirtualRequests, that enables you to require that all inbound requests be sent using SSL/TLS. When this property is set to *True,* the gateway accepts only those requests that are sent using SSL/TLS.

Configuring Synchronous Secondary Template Domains

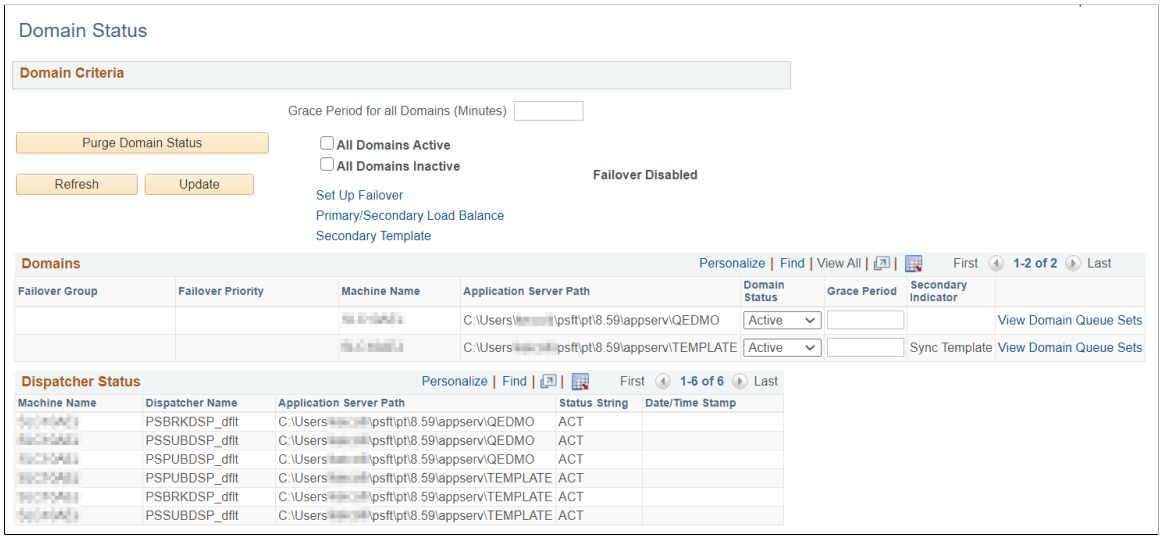
The first step to implementing load balancing using virtual applications server domains is to configure a synchronous secondary template domain.

You use PSADMIN to create synchronous secondary template domains, selecting the *IB Sync Secondary configuration template* option. Creating secondary templates is described elsewhere in the documentation.

See Creating Template Secondary Domains.

After you create a synchronous secondary template, the domain and its dispatcher processes appear on the Domain Status page in the Service Operations Monitor. To access the page, select **PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status.**

This example illustrates the Domain Status page when a synchronous template secondary is defined. The Domains grid shows the primary domain and the synchronous template secondary domain. The Dispatcher Status grid shows the dispatcher processes for both domains.

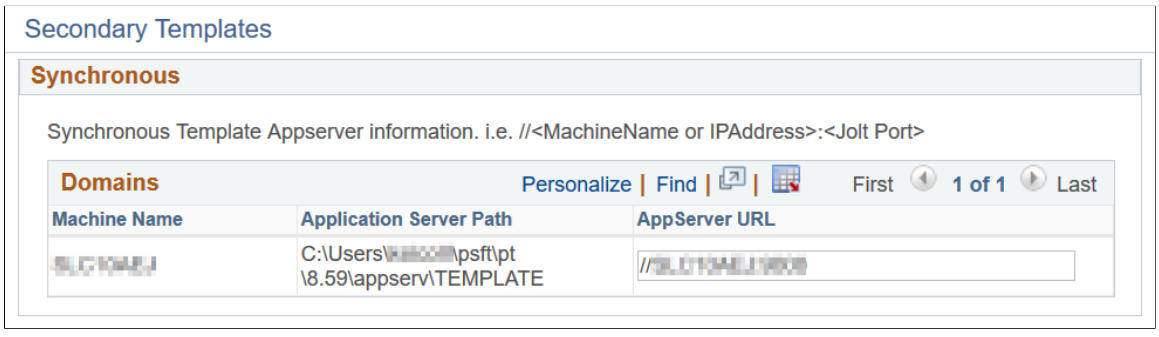


The Domains section of the page shows the primary domain as well as one synchronous secondary template domain. Note that the Secondary Indicator column for the synchronous secondary template reads *Sync Template* to identify it as a synchronous secondary template. The Dispatcher Status section displays the dispatcher processes for both the primary domain and the synchronous secondary template domain.

Defining Application Server URLs for Load Balancing

When you create a synchronous secondary template domain the system automatically registers the application URL in the Synchronous section of the Secondary Template page. To access the page, select **PeopleTools** > **Integration Broker** > **Service Operations Monitor** > **Administration** > **Domain Status** and click the **Secondary Templates** link.

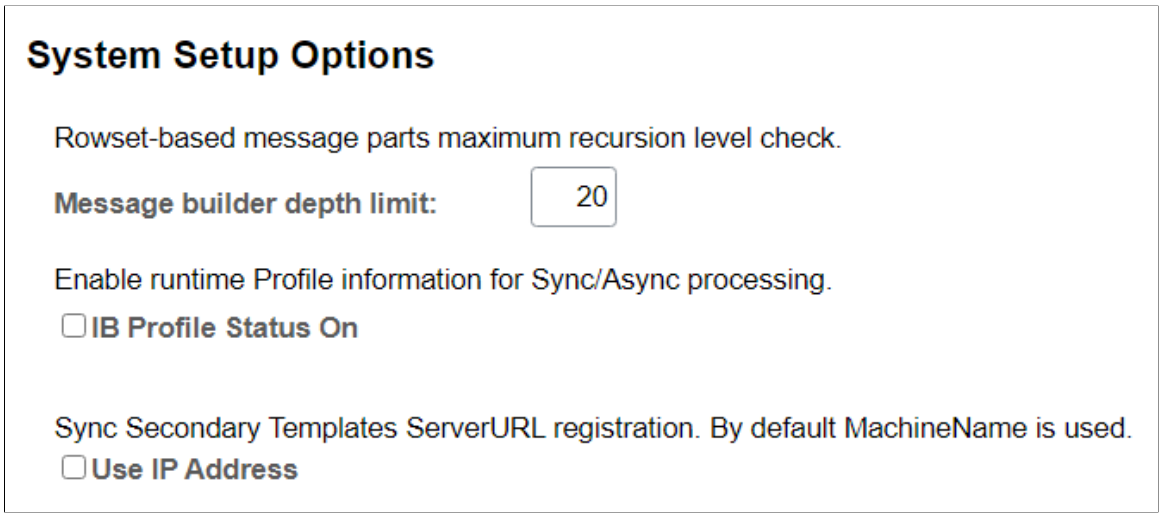
This example illustrates the Secondary Templates page. The system auto-populates the application server URL the synchronous secondary template in the **AppServer URL** field.



By default the system registers the URL in *<machine name>:<jolt port>* format. You can change the default so that the system populates the information using the IP address. The System Setup Options page (IB_SYSTEMSETUP) features an option, **Use IP Address**, that you can select so that the system uses the application server IP address.

To access the System Setup Options page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Broker Options.**

This example illustrates the System Setup Options page. Select the **Use IP Address** option to use the IP address for the application server URL.



To define the application server URL for a synchronous secondary template using the application server IP address:

1. Access the System Setup Options page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Broker Options.**.)

   Check the **Use IP Address** box.

2. Click the **Save** button.

If you use the System Setup Options page to change the URL format from <*machine name*>:<*jolt port*> to IP address, you must synchronize the virtual application server with the gateways defined for load balancing. If you have auto-synchronization set up the synchronization occurs at the interval set for synchronization. You can also perform a manual synchronization.

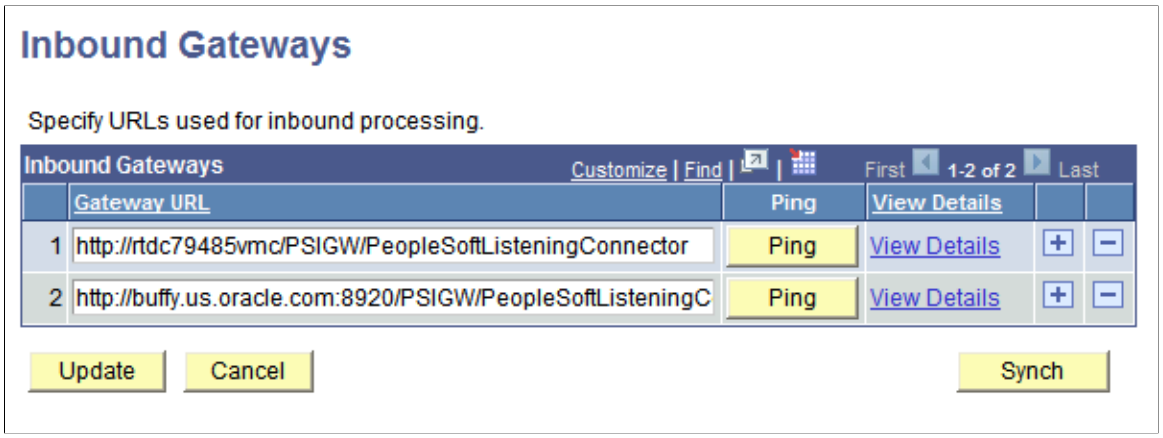
See Registering and Synchronizing Integration Gateways and Virtual Application Server Domains.

Defining Integration Gateways URLs for Inbound Processing

The system automatically uses the default local gateway for inbound request processing. You can add additional gateways to distribute processing tasks by using the Inbound Gateways page (IB_GATEWAY_SEC) in the Gateways component.

To access the Inbound Gateways page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Inbound Gateways** link.

This example illustrates the Inbound Gateways page. Use this page to define gateways for the processing of inbound integrations.



To integration gateway URLs for inbound processing.

1. Access the Inbound Gateways page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Inbound Gateways**and click the **Inbound Gateways** link).
2. Add row to the Inbound Gateways grid by clicking the add a row icon (+).
3. In the **Gateway URL** field enter a gateway URL in the following format:

   ```
   http://machinename:port/PSIGW/PeopleSoftListeningConnector
   ```

   In this case, *machinename:port* is the machine name and port, host name, or IP address of the web server hosting the gateway.
4. Click the *Update* button.

Related Links

Defining Virtual Server Nodes

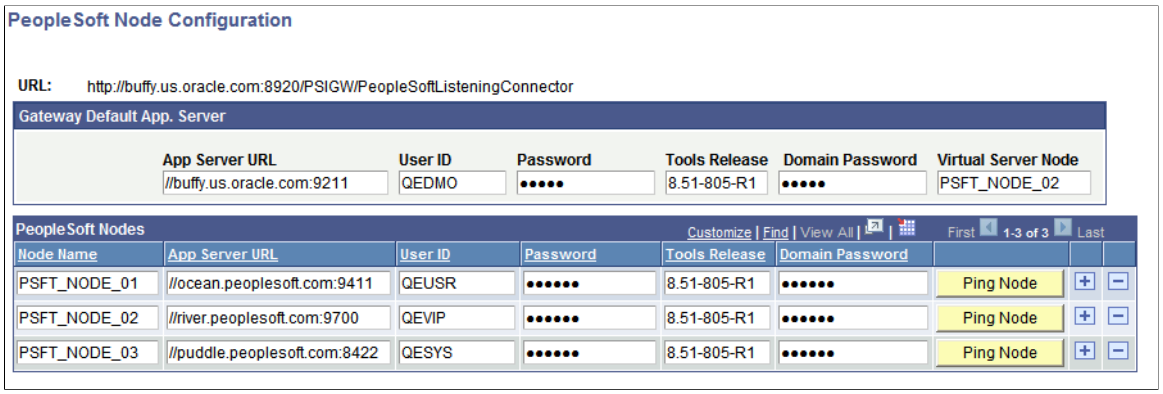
Defining Virtual Server Nodes

When an inbound request is sent without a destination node indicated, you can specify that a PeopleSoft node (as defined in the integration gateway properties file) process the request.

Note: Defining a virtual server node is optional.

The PeopleSoft Node Configuration page features a **Virtual Server Node** field. When you want a PeopleSoft node to process an inbound request that was sent without a destination node name, the integration system directs the request to the node specified in the field.

This example illustrates the PeopleSoft Node Configuration page. Use the **Virtual Node** field to define the node to process an inbound integration when no destination node is defined in the inbound request.



The example shows that the value *PSFT_NODE_02* is entered in the Virtual Server Node field. As a result, inbound requests that do not specify a destination node, are sent to the *PSFT_NODE__02* node.

You can also specify a virtual server node directly in the integrationGateway.properties file by setting the following property equal to the PeopleSoft node you want to handle the request:

```
ig.isc.virtualserverNode
```

If you specify this information directly in the integrationGateway.properties file, the JOLT connect information must adhere to the example provide in the example in the file. The order of properties must not deviate from the example provided in the properties file, shown here:

```
#ig.isc.virtualServerNode=VIRTUALSERVERNODE
#ig.isc.serverURL=//MYSERVER:9000
#ig.isc.userid=MYUSERID
#Use the supplied "Password Encryption Utility" to generate an encrypted
#password for the next entry.
#ig.isc.password=encrypted_password
#ig.isc.toolsRel=8.46
#Use the supplied "Password Encryption Utility" to generate an encrypted
#password for the next entry.
#ig.isc.domainConnectionPwd=encrypted_domain_password
```

Registering and Synchronizing Integration Gateways and Virtual Application Server Domains

This section discusses registering and synchronizing integration gateways and virtual application server domains.

Understanding Registering and Synchronizing Integration Gateways and Virtual Application Server Domains

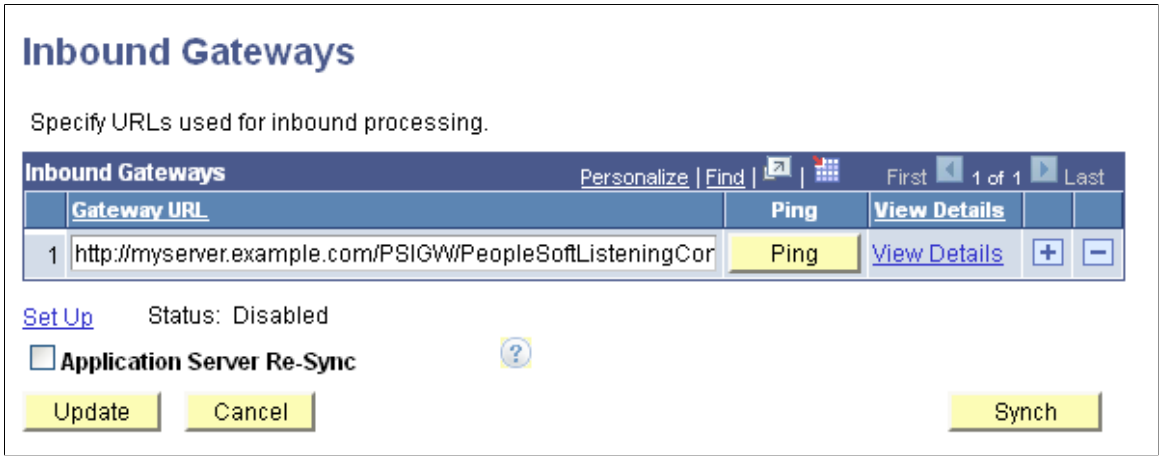
When you boot a virtual server it automatically registers itself with the gateways defined on the Inbound Gateways (IB_GATEWAY_SEC) page. Subsequently, when you shut down a virtual server it automatically de-registers itself.

There may be situations where the automatic registration process fails, a gateway gets restarted, and so on. You can manually register a virtual server domain or set up automatic synchronization at a specific regular interval.

Using the Inbound Gateways Page

The options for setting up manual or automatic synchronization are on the Inbound Gateways page. To access the Inbound Gateways page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Inbound Gateways** link.

This example illustrates the Inbound Gateways page. Use the **Ping** button or the **Set Up** link to register and synchronize virtual application servers with gateways.



The following fields and controls appear on the page:

Field or Control	***Description***
Gateway URL	Enter the integration gateway URL.
Ping	Click the button to ping the defined gateway.
View Details	Click the link to access the Gateway Server URLs page to view virtual server domains configured for the gateway. Using the Gateway Server URLs page is described later in this topic.
Set Up	Click the button to access the Auto Sync page to set up and enable auto-synchronization of the integration gateway and virtual application servers. Using the Auto Sync page is described later in this topic.
Status	This read-only field indicates the status of auto-synchronization. The values are: • *Disabled.* • *Enabled.*

Field or Control	*Description*
Application Server Re-Sync	This option is applicable only for situations where you are using a default application server configuration in the integration gateway (PeopleSoft Node Configuration page) and there are no virtual servers defined. If you set this option and the default application server goes down, the system synchronizes the gateway and the default application server so that when the application server is brought back up, the gateway again uses the application server without the need to recycle the gateway. Click the Set Up link to access the Auto Sync page to set up auto-synchronization of the default application server and the integration gateway.
Sync	Click the button to manually synchronize the integration gateway with virtual application servers.
Update	Click the button after performing a manual synchronization of the integration gateway and virtual application servers to complete the action.
Cancel	Click the button to exit the Inbound Gateways page without saving any changes.

Manually Registering and Synchronizing Virtual Server Domains with Gateways

To manually register and synchronize virtual server domains with gateways:

1. Access the Inbound Gateway page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Inbound Gateways** link).
2. Click the **Sync** button.
3. Click the **Update** button.

To set up the system to automatically register and synchronize virtual application servers with gateways use the Auto Sync page (IB_AUTOSYNC_SEC). To access the page, click the **Set Up** link on the Inbound Gateways page.

Note: When auto-synchronization is enabled, you may still use the Sync button to perform manual synchronization if needed.

This example illustrates the Auto Sync page.

Setting Up Auto-Registration and Auto-Synchronization of Virtual Application Servers with Gateways

To set up auto-registration and auto-synchronization of virtual application servers with gateways, at a minimum you must enable the Auto Sync feature and set the interval at which to perform the synchronization. You can also specify the email addresses of people to receive system-generated notifications in the event auto-synchronization fails or if there is a change to any of the set up parameters.

To enable auto-synchronization of virtual application server domains with gateways:

1. Access the Inbound Gateway page (**PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Inbound Gateways** link.)
2. Click the **Set Up** link.

   The Auto Sync page appears.
3. Check the **Enable Auto Sync** box.
4. In the **Auto Sync Time** field, enter the interval in minutes at which the system is to perform the synchronization.

   The default value is *5* minutes.
5. (Optional) In the **E-Mail To:** field enter the email addresses of people to receive system-generated email notification if auto-synchronization fails or if there is a change to any of the set up parameters. Separate multiple email address entries with a semi-colon.
6. (Optional) In the **E-Mail CC:** field enter the email addresses of persons to copy on system-generated email notifications of auto-synchronization failure or changes to the set up parameters. Separate multiple email address entries with a semi-colon.
7. Click the **OK** button.

   The Inbound Gateways page appears and the auto-sync status displays as *Enabled.*

Registering and Synchronizing Virtual Application Server Domains with Nodes

When the gateway is booted, upon initialization the system checks the parameter ig.isc.VirtualSynchronization in the integration gateway properties file. If set to *True,* the system looks at

all known nodes that are set to a specific PeopleTools release, for example PeopleTools 8.55, and makes a call to those endpoints to get the virtual server information.

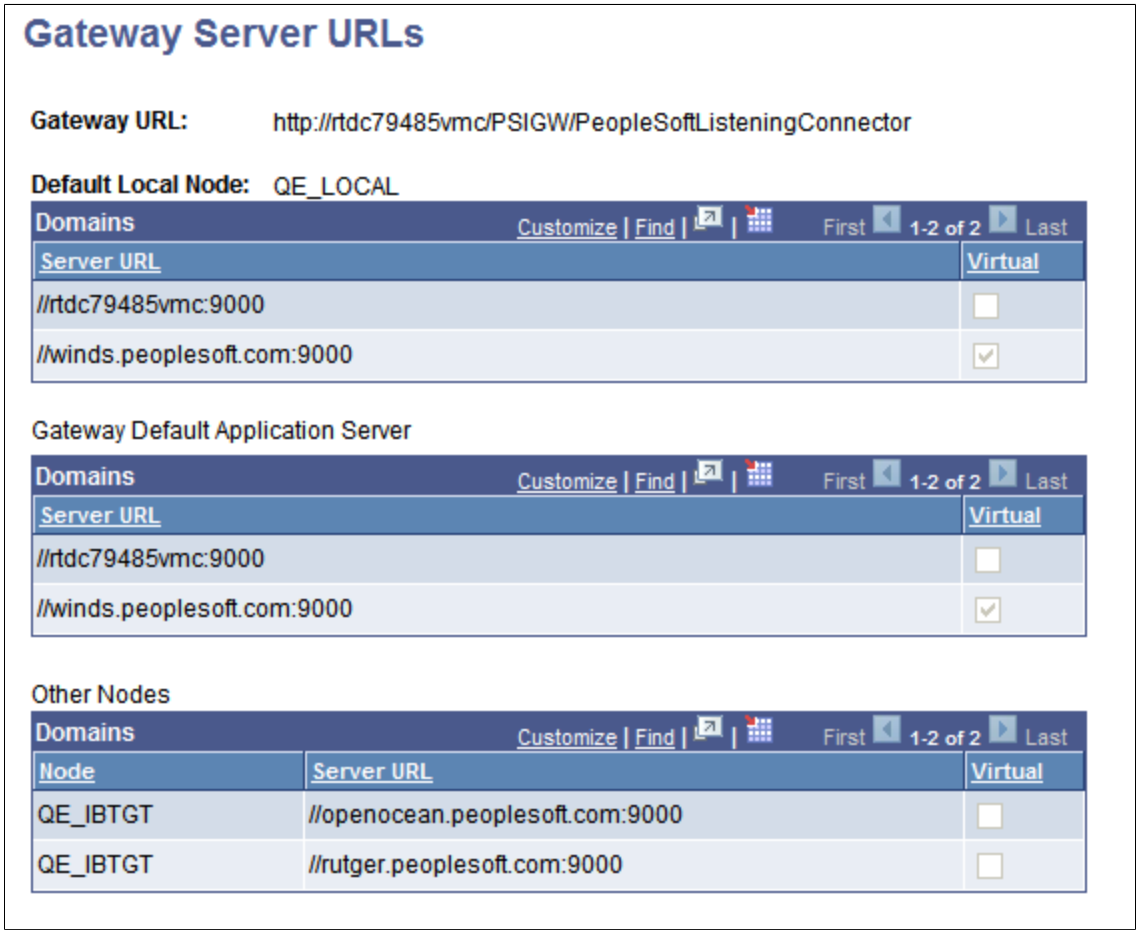
The ig.isc.VirtualSynchronization property appears in the Virtual Server section of the integration gateway properties file.

See Using the integrationGateway.properties File.

Viewing Virtual Application Server Domains Registered to Integration Gateways

The Gateway Server URLs page (IB_GWSERVER_SEC) lists the server URLs assigned to each node for a particular integration gateway. To access the Gateway Server URLs page, select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** > **Inbound Gateways** and click the **View Details** link for a gateway.

This example illustrates the Gateway Server URLs page. Virtual server URLs are identified on this page by a check in the Virtual column.



Enforcing Secure Inbound Requests

You can require that all inbound requests be sent using Secure Socket Layer security/Transport Layer Security (SSL/TLS).

To do so, set the following property in the integrationGateway.properties file to *True.*

```
ig.SecureVirtualRequests=True
```

This property is located in the Virtual Server Section of the properties file.

When this property is set to *True,* the gateway accepts only those requests that are sent using SSL/TLS.

Auto-Synchronizing Default Application Servers and Integration Gateways

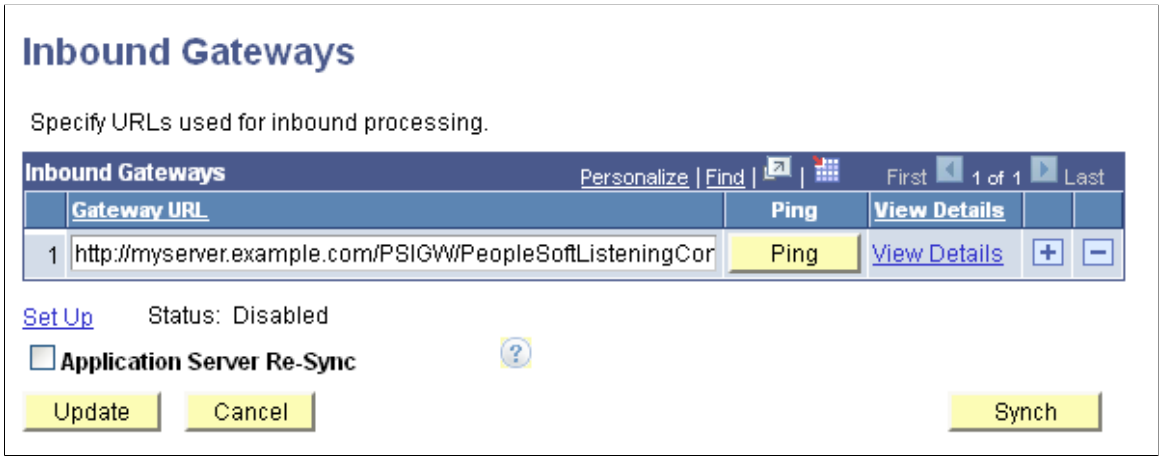
Use the Inbound Gateways page and the Auto Sync page to enable and set up the auto-synchronization of a default application server and the integration gateway should the application server go down.

This option is applicable only for situations where you are using a default application server configuration in the integration gateway (as defined in the Nodes – PeopleSoft Node Configuration page) and there are no virtual servers defined in the Inbound Gateways page.

When this feature is enabled and the default application server goes down, the system synchronizes the gateway and the default application server so that when the application server is brought back up, the gateway again uses the application server without the need to recycle the gateway.

To access the Inbound Gateways page select **PeopleTools** > **Integration Broker** > **Configuration** > **Integration Gateways** and click the **Inbound Gateways** link.

This example illustrates the Inbound Gateways page. Use the page to enable the application server re-synchronization feature.



The fields and controls on the Inbound Gateways page are described in the Implementing Inbound Request Load Balancing Using Virtual Application Server Domains topic.

To access the Auto Sync page, click the **Set Up** link on the Inbound Gateways page.

This example illustrates the Auto Sync page. Use the page to set up the automatic synchronization feature.

Using WS-Reliable Messaging

This section discusses how to:

- Use WS-Reliable Messaging on outbound service operations.
- Use WS-Reliable Messaging on inbound service operations.

Understanding WS-Reliable Messaging

Web Service (WS) Reliable Messaging is a protocol that allows SOAP messages to be delivered reliably between distributed applications in the presence of failures at the software component, system, or network level.

Using WS-Reliable Messaging on Outbound Service Operations

This section discusses using WS-Reliable Messaging on outbound service operations.

Understanding Using WS-Reliable Messaging on Outbound Service Operations

For outbound service operations, PeopleSoft supports WS-Reliable Messaging for asynchronous transactions.

Outbound service operations only support a single content section, which will correspond to a single message with a WS-Reliable Messaging sequence header block on the wire. Multiple contents sections, and therefore multiple runtime sequence messages, are not supported.

When using reliable messaging, message bodies should not be SOAP wrapped in the application server. The gateway builds a SOAP envelope as a by-product of the WS-Reliable Messaging processing.

During a successful invocation, three WS-Reliable Messaging messages are sent out on the wire: a CreateSequence, a message containing the message data as well as a Sequence header block, and a TerminateSequence message. If errors are seen during the transmission of a WS-Reliable Messaging message sequence, the gateway does not auto-recover. Errors are propagated back to the application server and it is the responsibility of the developer to handle further processing.

Enabling WS-Reliable Messaging on Outbound Service Operations

To use WS-Reliable Messaging on outbound transactions, use the HTTP target connector. The HTTP target connector features a gateway–level property called *WS-RM*. When you set the value of the property to *Y*, the transaction uses the WS-Reliable Messaging protocol for service operation delivery. Like any other HTTP target connector property, you can override the setting on the service operation routing or by using PeopleCode.

Setting the Value of AcksTo Endpoints in CreateSequence Messages

The integrationGateway.properties file features the following property that enables you to set the value of the AcksTo endpoint in CreateSequence messages:

```
ig.WSRM.CreateSequenceAcksTo=
```

The property is located in the WS Reliable Messaging section of the gateway properties file.

Note that the system expects to receive responses on the back channel, so the value you set for this property has no bearing on how the Integration Broker processes WS-Reliable Messaging response messages.

Related Links

Editing Connector Properties
Using the integrationGateway.properties File

Using WS-Reliable Messaging on Inbound Service Operations

For inbound transactions, Integration Broker accepts the WS-Reliable Messaging protocol when posted into the PeopleSoft Service or HTTP listening connectors. The feature is triggered by the presence of the WS-Reliable Messaging namespace and headers in received SOAP messages.

Note that the WSDL does not reflect the use of WS-Reliable messaging.

For inbound service operations, PeopleSoft supports WS-Reliable Messaging for asynchronous transactions only. If an integration partner sends a synchronous transaction to Integration Broker using WS-Reliable messaging, an error message is returned to the sending system.

All inbound WS-Reliable Messaging communication between the sender and Integration Broker occurs in a single HTTP request-response transaction. The sender transmits a WS-Reliable Messaging message, Integration Broker receives and processes it, and then returns a WS-Reliable Messaging message in the HTTP response. Integration Broker ignores the value of AcksTo in CreateSequence messages.

Unlike outbound transactions using WS-Reliable Messaging, multiple Sequence messages are supported for inbound transactions.

Using the Bulk Load Handler for Large Message Subscriptions

PeopleSoft Integration Broker provides a bulk load handler type that serves as a bulk loader to insert data. This handler is available when working with asynchronous one-way service operation types.

See “Implementing Handlers Using Bulk Load Processing” (Integration Broker).

Managing Pub/Sub Process Handler Performance

This section discusses how to:

- Enable the serial recycling of pub/sub process handlers.
- Recycle pub/sub process handlers based on process memory growth.

Enabling Serial Recycling of Pub/Sub Process Handlers

When serial recycle for pub/sub process handlers is enabled, the system recycles process handlers (within a group) on a serial basis–one after another–to allow processing to continue uninterrupted.

If serial recycling is disabled, all pub/sub process handlers recycle at once, which can cause throughput to come to a standstill.

By default the serial recycling of pub/sub process handlers is enabled.

Serial recycling uses the following parameters that you set in the psappsrv.cfg file in the Settings for PUB/SUB Servers section:

Field or Control	***Description***
Serial Recycle	To enable serial recycling enter *Y*. To disable serial recycling enter *N*. The default value is *Y*.
Serial Recycle Time	Specifies the maximum interval in seconds at which the system recycles a process. The minimum valid recycle time is *60* seconds The default value is *60* seconds.

To enable serial recycling, uncomment the parameters in the psappsrv.cfg file and set the appropriate values. After you have made your changes, save the file and reboot the application server.

To disable serial recycling, comment out the parameters, save the file and reboot the application server.

Recycling Pub/Sub Process Handlers Based on Process Memory Growth

PeopleSoft Integration Broker enables you to recycle pub/sub process handlers based on memory growth in cache.

You use the **Percentage of Memory Growth** parameter in the psappsrv.cfg file to specify that the system recycle pub/sub process handlers when memory has grown by a value you specify. The system checks to see if the percentage criterion is met after processing a specified number of requests.

By default the **Percentage of Memory Growth** parameter is disabled.

This feature uses the following parameters that you set in the psappsrv.cfg file in the Settings for PUB/SUB Servers section:

Field or Control	***Description***
Percentage of Memory Growth	Specifies the percentage growth of memory in cache at which the system recycles pub/sub process handlers. The default value is *20* percent.
Interval Parameter	Determines the number of requests for the system to process before checking the percent memory growth in cache. The default value is *100.*

To enable process handler recycling based on memory growth, uncomment the parameters in the psappsrv.cfg file and set the appropriate values. After you have made your changes, save the file and reboot the application server.

To disable process handler recycling based on memory growth, comment out the parameters, save the file and reboot the application server.

Chapter 16

Using the Delivered Listening Connectors and Target Connectors

Understanding Using the Connector Examples

This topic presents examples of how to use the connectors delivered with PeopleSoft Integration Broker.

The intention of the examples provided in this topic is to provide a starting point for exploring how the connectors work. The examples are designed to be simple and require the minimum set up and configuration necessary to invoke them.

If you try these examples and choose to cut the code samples provided in this document and paste them into PeopleSoft Application Designer, the PeopleSoft Pure Internet Architecture, or text or XML editors, verify that single or double quotation marks are pasted into these mediums as straight quotes. Slanted or curly quotes will cause the code samples to fail.

Related Links

Setting Up Metadata
Example 1: Using the PeopleSoft Connectors
Example 2: Using the HTTP Connectors
Example 3: Using the PeopleSoft 8.1 Connectors
Example 4: Using the JMS Connectors
Example 5: Using the AS2 Connectors
Example 6: Using the FTP Target Connector
Example 7: Using the SFTP Target Connector
Example 8: Using the SMTP Target Connector

Prerequisites

To use this topic, you should have basic experience in using PeopleSoft Integration Broker. There is little background information presented in this topic and many of the basic steps involved in creating integrations are presented in general terms (for example, “create a new Service/Service Operation.”) Please refer to the appropriate information in the product documentation for information on how to complete basic tasks.

Setting Up Metadata

This section discusses how to set up metadata for the examples presented in this topic and discusses how to:

- Create queues, request messages, response messages, services, service operations and pages.
- Create nodes and routing definitions.
- Create a test record and page.
- Set up integration gateway logging.

Understanding Setting Up Metadata

Before you use the examples in this topic you must set up metadata as described in this section.

Note: The examples presented in this topic demonstrate the use of one type of connector at a time. The examples share the same basic definitions for the service operation, request message, response message, routings, and the test page. As a result, you should attempt to run only one example at a time, since the underlying metadata and objects are shared.

The exact requirements for setting up the listening and target connectors do differ somewhat, but since the differences are fairly minor the steps are combined in this section.

Prerequisites for Setting Up Metadata

Before you begin the set up data for the examples configure and start the integration gateway.

Creating Services, Service Operations, Queues, and Messages

This section describes creating services, service operations, queues, and request and response messages for use in running the connector examples presented in this topic.

Unless otherwise noted, use the appropriate PeopleSoft Pure Internet Architecture pages to complete these tasks.

To create services, service operations, queues, and messages:

1. Create a new request message.

   Create a Nonrowset-based message with message name as *EXAMPLE_REQUEST_MSG* and message version as *VERSION_1*.

2. Create a new response message.

   Create a Nonrowset-based message with message name as *EXAMPLE_RESPONSE_MSG* and message version as *VERSION_1*.

3. Create a new Service

   Name the service *EXAMPLE_SERVICE.*

4. Create new synchronous Service operation.

   a. Add a service operation of type synchronous to the *EXAMPLE_SERVICE* service and name it *EXAMPLE_SERVICE_OPR*.

b. Complete the field definitions for service operation as follows:

Field	***Value***
Operation Description	Test service operation
Request Message.Version	*EXAMPLE_REQUEST_MSG.VERSION_1*
Response Message Name.Version	*EXAMPLE_RESPONSE_MSG.VERSION_1*

c. Configure the Service Operation Security for this service operation.

5. Create a new asynchronous Service Operation

a. Add a service operation of type Asynchronous — one way to the EXAMPLE_SERVICE and name it *EXAMPLE_SERVICE_ASYNC_OPR*

b. Complete the field definitions for the service operation as follows:

Field	***Value***
Operation Description	Test service operation
Request Message.Version	*EXAMPLE_REQUEST_MSG.VERSION_1*
Queue Name	EXAMPLE_QUEUE

c. Configure the Service Operation Security for this service operation.

6. Create a new queue.

a. Name the queue *EXAMPLE_QUEUE*

b. Verify that the **Queue Status** is set to **Run**.

c. Use the Integration Broker Service Operations Monitor Administration to verify that the *EXAMPLE_QUEUE* is running.

Creating the Test Record and Page

This section discusses how to use PeopleSoft Application Designer to:

- Create a test record.
- Create a test page.

Creating the Test Record

You must create a work record that will be used on the Test Page.

Create a new record:

1. Insert the character field *TEST* into the record.
2. Select **Derived/Work** as the **Record Type**.
3. Save the record as *EXAMPLE_WORKREC*.

Creating the Test Page

You must create a test page. This page will be used in some of the target connector examples.

Create a new page with a single push button on it:

1. Create the page.
2. Add a push button with the following properties:

Property	***Value***
Destination	**PeopleCode Command**
RecordName	*EXAMPLE_WORKREC*
Field Name	*TEST*

3. Re-size the button and label it *Test target connector*.
4. Save the page as *EXAMPLE_PAGE*.
5. Add the page to a component. This may be an existing component or a new one. Ensure that the security settings for the component allow the new page to be accessed.

Creating Nodes and Routing Definitions

Use the PeopleSoft Pure Internet Architecture to complete the following tasks.

Creating Source Nodes and Inbound Routing for Service Operations

You must create a node that will be the source of all requests to the listening connectors.

To create a source node and a inbound routing:

1. Add a new node called *SOURCENODE*. Enter in appropriate values for the description and the default user ID. Verify that the **Active Node** check box has been selected. Save this node.
2. Add a new inbound routing to the *EXAMPLE_SERVICE_OPR* service operation and name it EXAMPLE_SERVICE_IN_RTN.

a. Set the **Sender Node** field value to *SOURCENODE* and the **Receiver Node** field value to the local node's value.

b. Check the *Active* check-box for routing .

c. Set the **Logging Details** field value to *Header and Detail* .

d. Save the routing.

Adding Target Nodes and Outbound Routing

You must create a target node and an outbound routing for all outgoing requests for the target connectors.

To add a target node and an outbound routing:

1. Add a new node called *TARGETNODE*. Enter in the appropriate values for the description and default user ID. Verify that the **Active Node** check box has been selected. Save this node.
2. Add a new outbound routing to the *EXAMPLE_SERVICE_OPR* service operation and name it *EXAMPLE_SERVICE_OUT_RTN.*

   a. Set the **Sender Node** field value to the local node's value and the **Receiver Node** field value to *TARGETNODE.*

   b. Verify that the **Status** is set to **Active**.

   c. Verify that **Logging Details** field value is set to *Header and Detail*.

   d. Save the routing.

3. Add a new outbound routing to the service operation *EXAMPLE_SERVICE_OPR_ASYNC* and name it *EXAMPLE_SERVICE_ASYNC_RTN.*

   a. Set the **Sender Node** field value to the local node's value and the **Receiver Node** field value to *TARGETNODE.*

   b. Verify that the **Status** is set to **Active**.

   c. Verify that **Logging Details** field value is set to *Header and Detail*.

   d. Save the routing.

Setting Up Integration Gateway Logging

The integration gateway has message and error logging capabilities. If problems arise while trying the examples, these logs can be invaluable in determining where problems are occurring.

See “Managing Integration Gateway Message and Error Logging” (Integration Broker).

Example 1: Using the PeopleSoft Connectors

This section discusses using the PeopleSoft listening and PeopleSoft target connectors.

Understanding the PeopleSoft Connector Examples

The example provided for using the PeopleSoft target connector demonstrates using the connector to invoke a synchronous service operation between two PeopleSoft nodes.

The example provided for using the PeopleSoft listening connector demonstrates using Send Master to invoke a service operation into the local system for processing.

Prerequisites

To use the PeopleSoft target connector example you must have a second PeopleSoft 8.55 system. You must have the application server, the PeopleSoft Pure Internet Architecture and the Integration Gateway configured and running.

Note: In this section, the current PeopleSoft system is referred to as the *originating* system, and the second PeopleSoft system is called the *destination* system.

Using the PeopleSoft Target Connector

This section provides an example of using the PeopleSoft target connector and describes how to:

- Set up data on the originating system.
- Set up data on the destination system.
- Test the PeopleSoft target connector.

Setting Up Data on the Originating System

To set up data on the originating system:

1. In PeopleSoft Application Designer, open the EXAMPLE_WORKREC record. Add the following PeopleCode to the FieldChange event for the TEST field:

```
&msg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);

&xmldata = "<?xml version='1.0'?><ConnectorTest/>";

/* create an XmlDoc */
&xmlDoc = CreateXmlDoc(&xmldata);
&rootNode = &xmlDoc.documentelement;
&descNode = &rootNode.addelement("PSFTtest");
&descNode.nodevalue = "This is a test message.";

/* put the XML in the message */
&msg.setxmldoc(&xmlDoc);

/* send the request */
&response = %IntBroker.SyncRequest(&msg);
```



```
/* and echo it back to the user */
&xmlDoc = &response.getxmldoc();
MessageBox(0, "", 0, 0, &xmlDoc.genxmlstring());
```

2. In the PeopleSoft Pure Internet Architecture, open the node definition for TARGETNODE. Set the ConnectorID to PSFTTARGET.

3. In the Integration Properties for the gateway, add a new entry for TARGETNODE along with the appropriate values.

```
ig.isc.TARGETNODE.serverURL=//<machinename>:<port>
ig.isc.TARGETNODE.userid=<userid>
ig.isc.TARGETNODE.password=<password>
ig.isc.TARGETNODE.toolsRel=<toolsRelease>
```

Setting Up Data on the Destination System

To set up data on the destination system:

1. Follow the steps outlined in the section Setting Up Metadata to add the following to the destination system:

   a. The *EXAMPLE_QUEUE* queue.

   b. The *EXAMPLE_REQUEST_MSG* message.

   c. The *EXAMPLE_RESPONSE_MSG* message.

   d. The *EXAMPLE_SERVICE* service.

   e. The *EXAMPLE_SERVICE_OPR* synchronous service operation.

2. Add a node entry for the originating system. Ensure that the Single Signon security is configured so that the destination system accepts authentication tokens from the originating system.

3. Add a new inbound synchronous routing between the originating system and the destination for the *EXAMPLE_SERVICE_OPR* service operation.

4. In the PeopleSoft Pure Internet Architecture, for service operation *EXAMPLE_SERVICE_OPR* add a handler of type *OnRequest* with implementation type *App Class.* Create a handler application class based on the IRequestHandler interface, and for the method OnRequest add the following PeopleCode:

```
Local XmlDoc &xmldoc;
Local File &theFile;
Local XmlNode &rootNode, &descNode;
Local Message &response;
Local string &xmldata;

/* get the body of the incoming message */
&xmldoc = &_MSG.GetXmlDoc();

/* and write it out to a file */
&theFile = GetFile("ARequest.txt", "W", "UTF8");
&theFile.WriteString(&xmldoc.GenXmlString());
&theFile.Close();

/* create the response message */
&response = CreateMessage(Operation.EXAMPLE_SERVICE_OPR, %IntBroker_Respons⇒
```



```
e);

   /* create the body for the response message */
   &xmldata = "<?xml version='1.0'?><ConnectorTest/>";
   &xmldoc = CreateXmlDoc(&xmldata);
   &rootNode = &xmldoc.DocumentElement;
   &descNode = &rootNode.AddElement("ResponseMessage");
   &descNode.NodeValue = "This was generated in the OnRequest event.";

   /* add the body to the message */
   &response.SetXmlDoc(&xmldoc);

   /* and return the response message */

   Return &response;
```

Testing the PeopleSoft Target Connector

To test the PeopleSoft target connector:

1. In the PeopleSoft Pure Internet Architecture, open the EXAMPLE_PAGE page and click the **Test** button. The response message will be displayed in a message box.
2. On the destination system, open Service Operations Monitor to view the details of the received message. Open the text file created by the OnRequest PeopleCode to view the details of service operation request received.

Using the PeopleSoft Listening Connector

This section provides an example for testing the PeopleSoft listening connector.

Testing the PeopleSoft Listening Connector

To test the PeopleSoft listening connector:

1. In the PeopleSoft Pure Internet Architecture, open the *EXAMPLE_SERVICE_OPR* service operation and add a handler of type OnRequest with implementation type App class. The OnRequest method of App class should have following PeopleCode:

```
   Local XmlDoc &xmldoc;
   Local File &theFile;
   Local XmlNode &rootNode, &descNode;
   Local Message &response;
   Local string &xmldata;

   /* get the body of the incoming message */
   &xmldoc = &_MSG.GetXmlDoc();

   /* and write it out to a file */
   &theFile = GetFile("HttpRequest.txt", "W", "UTF8");
   &theFile.WriteString(&xmldoc.GenXmlString());
   &theFile.Close();

   /* create the response message */
   &response = CreateMessage(Operation.EXAMPLE_SERVICE_OPR, %IntBroker_Respons⇒

e);

   /* create the body for the response message */
   &xmldata = "<?xml version='1.0'?><ConnectorTest/>";
   &xmldoc = CreateXmlDoc(&xmldata);
   &rootNode = &xmldoc.DocumentElement;
```



```
&descNode = &rootNode.AddElement("ResponseMessage");
&descNode.NodeValue = "This was generated in the OnRequest event.";

/* add the body to the message */
&response.SetXmlDoc(&xmldoc);

/* and return the response message */

Return &response;
```

2. Start Send Master and create an **8.48 Integration Broker (MIME)** project.
3. In the **URL** field enter the address of the PeopleSoft listening connector:

   ```
   http://your_server_name/PSIGW/PeopleSoftListeningConnector
   ```

   Replacing <*your_server_name*> with the details of the server where the gateway is running. For example:

   ```
   http://machine1234/PSIGW/PeopleSoftListeningConnector
   ```

4. In the **Requesting Node** field, enter *SOURCENODE*.
5. In the **Ext. Operation name** field, enter *EXAMPLE_SERVICE_OPR.v1*.
6. From the **Operation type** list, select **Sync**.
7. Click the Input File tab and enter the following XML:

   ```
   <?xml version="1.0"?><Test>Data</Test>
   ```

8. Click the **Post** button.

   The response from the server displays in the Output Information section. Note that this is a MIME response; look near the end to find the response XML generated by the OnRequest PeopleCode. Open the text file created by the OnRequest method of application class to view the body of the request message.

Example 2: Using the HTTP Connectors

This section discusses how to:

- Use the HTTP listening connector.
- Use the HTTP target connector.

Prerequisites

When using the examples for using the HTTP target connector, an HTTP server is needed to receive the HTTP request and to return a response. If using the SOAP example, the HTTP server must be able to process SOAP messages.

Using the HTTP Listening Connector

This section provides examples of how to set credentials for HTTP requests coming into the integration gateway, and discusses how to:

- Set credentials in message bodies.
- Set credentials in HTTP headers.
- Set credentials in query strings.
- Set credentials in SOAP-specific HTTP headers.

Setting Up for Using the HTTP Listening Connector Examples

In the PeopleSoft Pure Internet Architecture, for service operation *EXAMPLE_SERVICE_OPR* add a handler of type OnRequest with implementation type application Class. Create a handler application class based on the IRequestHandler interface, and for the method OnRequest add following PeopleCode

```
Local XmlDoc &xmldoc;
Local File &theFile;
Local XmlNode &rootNode, &descNode;
Local Message &response;
Local string &xmldata;

/* get the body of the incoming message */
&xmldoc = &_MSG.GetXmlDoc();

/* and write it out to a file */
&theFile = GetFile("HttpRequest.txt", "W", "UTF8");
&theFile.WriteString(&xmldoc.GenXmlString());
&theFile.Close();

/* create the response message */
&response = CreateMessage(Operation.EXAMPLE_SERVICE_OPR,
  %IntBroker_Response);

/* create the body for the response message */
&xmldata = "<?xml version='1.0'?><ConnectorTest/>";
&xmldoc = CreateXmlDoc(&xmldata);
&rootNode = &xmldoc.DocumentElement;
&descNode = &rootNode.AddElement("ResponseMessage");
&descNode.NodeValue = "This was generated in the OnRequest event.";

/* add the body to the message */
&response.SetXmlDoc(&xmldoc);

/* and return the response message */
Return &response;
```

Setting Credentials in the Message Body

To set HTTP request credentials in the message body:

1. Start Send Master, and create a new Input File project.
2. In the URL field enter:

```
http://<your_server_name>/PSIGW/HttpListeningConnector
```


Replace <*your_server_name*> with the details of the server where the integration gateway is running. For example:

```
http://machine1234/PSIGW/HttpListeningConnector
```

3. In the Input section, paste the following XML. Notice that the service operation name and requesting node are present in the XML.

```
<?xml version="1.0"?>
<IBRequest>
  <ExternalOperationName>EXAMPLE_SERVICE_OPR.v1</ExternalOperation
   Name>
  <From>
    <RequestingNode>SOURCENODE</RequestingNode>
 </From>
 <ContentSections>
   <ContentSection>
     <Data>
       <![CDATA[<?xml version="1.0"?><ConnectorTest>
       Testing the HTTPListeningConnector. Message body.
       </ConnectorTest>]]>
     </Data>
   </ContentSection>
 </ContentSections>
</IBRequest>
```

4. Click the **Post** button to invoke service operation on the integration gateway.

5. Check the Output section for the response. Compare the response with the XML created in the handler application class. Also check the HttpRequest.txt file created by the OnRequest PeopleCode to see the body of the request message received by the application server.

Setting Credentials in HTTP Headers

To set HTTP request credentials in the HTTP header:

1. Start Send Master, and create a new Input File project.

2. In the URL field enter:

```
http://<your_server_name>/PSIGW/HttpListeningConnector
```

Replace <*your_server_name*> with the details of the server where the integration gateway is running. For example:

```
http://machine1234/PSIGW/HttpListeningConnector
```

3. In the **Headers** field enter the following:

```
OperationName:EXAMPLE_SERVICE_OPR.v1
From:SOURCENODE
```

4. In the Input section, paste the following:

```
<?xml version="1.0"?>
<ConnectorTest>
Testing the HTTPListeningConnector. HTTP Header.
</ConnectorTest>
```

5. Click the **Post** button to sent the message to the integration gateway.

6. Check the Output section for the response. Compare the response with the XML created in the handler application class. Also check the HttpRequest.txt file created by the OnRequest PeopleCode to see the body of the request message received by the application server.

Setting Credentials in Query Strings

To set HTTP request credentials in a query string:

1. Start Send Master, and create a new Input File project.
2. In the **URL** field enter:

   ```
   http://your_server_name/PSIGW/HttpListeningConnector?&Operation=
   EXAMPLE_SERVICE_OPR.v1&From=SOURCENODE
   ```

   Replace <your_server_name> with the details of the server where the integration gateway is running. For example:

   ```
   http://machine1234/PSIGW/HttpListeningConnector?&Operation=
   EXAMPLE_SERVICE_OPR.VERSION_1&From=SOURCENODE
   ```

3. In the Input section, paste the following:

   ```
   <?xml version="1.0"?>
   <ConnectorTest>
   Testing the HTTPListeningConnector. Query String.
   </ConnectorTest>
   ```

4. Click the **Post** button to invoke service operation on the integration gateway.
5. Check the Output section for the response. Compare the response with the XML created in the handler application class. Also check the HttpRequest.txt file created by the OnRequest PeopleCode to see the body of the request message received by the application server.

Setting Credentials in SOAP-Specific HTTP Headers

To set HTTP request credentials in a SOAP-specific HTTP header:

1. Start Send Master, and create a new Input File project.
2. In the URL field enter:

   ```
   http://your_server_name/PSIGW/HttpListeningConnector
   ```

   Replacing <your_server_name> with the details of the server where the gateway is running. For example:

   ```
   http://machine1234/PSIGW/HttpListeningConnector
   ```

3. In the **Header** field, add the following:

   ```
   SOAPAction: http://example.com/EXAMPLE_SERVICE_OPR.v1/SOURCENODE//
   ```

4. In the Input section, paste the following:

   ```
   <?xml version="1.0"?>
   <SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.encoding/
    "xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
       <SOAP-ENV:Body>
         <ConnectorTest>
            <Text>
   ```



```
            Testing the HTTPListeningConnector. SOAP Message.
          </Text>
        </ConnectorTest>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5. Click the **Post** button to invoke service operation on the integration gateway.

6. Check the Output section for the response. Compare the response with the XML created in the handler application class; that XML will be returned wrapped in a SOAP envelope. Also check the HttpRequest.txt file created by the OnRequest PeopleCode to see the body of the request message received by the application server.

Using the HTTP Target Connector

This section provides examples of using the HTTP target connector and discusses how use the connector to:

- Send standard HTTP requests.
- Send SOAP messages in HTTP requests.

Sending Standard HTTP Requests

To send a standard HTTP request:

1. In PeopleSoft Application Designer, open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the *TEST* field.

```
&msg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);

&xmldata = "<?xml version='1.0'?><ConnectorTest/>";

/* create an XmlDoc */
&xmlDoc = CreateXmlDoc(&xmldata);
&rootNode = &xmlDoc.documentelement;
&descNode = &rootNode.addelement("HTTPtest");
&descNode.nodevalue = "This will be sent to an HTTP server.";

/* put the XML in the message */
&msg.setxmldoc(&xmlDoc);

/* send the request */
&response = %IntBroker.SyncRequest(&msg);

/* and echo it back to the user */
&xmlDoc = &response.getxmldoc();
MessageBox(0, "", 0, 0, &xmlDoc.genxmlstring());
```

   Note that this code assumes that the response from the server is properly formatted XML.

2. In the PeopleSoft Pure Internet Architecture, open the node definition for TARGETNODE. Set the **Connector ID** to *HTTPTARGET*. Set the **URL** property value to the address of the HTTP server that will process the request.

3. Open the *EXAMPLE_PAGE* page, and click on the **Test** button. The HTTP response will be displayed in the resulting message box.

Sending SOAP Messages in HTTP Requests

To send a SOAP message in an HTTP request:

1. In PeopleSoft Application Designer, open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the *TEST* field.

```
&msg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);

/* create a SOAP document */
&soapReq = CreateSOAPDoc();

&soapReq.AddMethod("TestNode", 1);
&soapReq.AddParm("Text", "This is a SOAP request.");

/* put the XML in the message */
&msg.setxmldoc(&soapReq.xmlDoc);

/* send the request */
&response = %IntBroker.SyncRequest(&msg);

/* and echo it back to the user */
&xmlDoc = &response.getxmldoc();
MessageBox(0, "", 0, 0, &xmlDoc.genxmlstring());
```

2. In the PeopleSoft Pure Internet Architecture, open the node definition for *TARGETNODE*.
   a. On the Node Definitions-Connectors tab, set the **Connector ID** to *HTTPTARGET*.
   b. Set the **URL** property value to the address of the HTTP server that will process the request.
3. Open the *EXAMPLE_PAGE* page, and click on the **Test** button. The HTTP response will be displayed in the resulting message box.

Example 3: Using the PeopleSoft 8.1 Connectors

The examples provided in this section demonstrate sending a rowset-based asynchronous message between a PeopleSoft 8.55 node and a PeopleSoft 8.1 node.

Understanding the PeopleSoft 8.1 Connectors Examples

When sending a message from a PeopleSoft 8.55 system to a PeopleSoft 8.1 system, you will use the PeopleSoft 8.1 target connector. You will also use PeopleCode, as well as the example page and work record that you created using the information in the setup section at the beginning of this topic.

When sending a message from a PeopleSoft 8.1 system to a PeopleSoft 8.4 system, you will use the PeopleSoft 8.1 listening connector. You will also use the test message functionality in PeopleSoft Application Designer.

Setting Up Data for the PeopleSoft 8.1 Connectors Examples

This section describes setting up data for using the PeopleSoft 8.1 connector examples.

Setting Up Data on the PeopleSoft 8.55 System

To set up data on the PeopleSoft 8.55 system:

1. In PeopleSoft Application Designer, create a new field called *EXAMPLE_CHAR*. This should be a mixed-case character field of size *20*.
2. Create a new record.
   a. Name the record *EXAMPLE_REC*.
   b. Add the *EXAMPLE_CHAR* field to this record, set it as the key, and save the definition.
   c. Build the physical table for this record.
3. In the PeopleSoft Pure Internet Architecture, create a new message called *EXAMPLE_PSFT_MSG* with the version set to *VERSION_1*.
   a. Select the message type to be Rowset— Based.
   b. Add the *EXAMPLE_REC* record as the root record of this message.
4. Add a new node, using the node name of the PeopleSoft 8.1 system. Verify that the **Active Node** box is checked, and save the record.
5. Open the **EXAMPLE_PAGE** page and add an **EditBox** to the page, setting the following properties:

Property	***Value***
Record name	*EXAMPLE_REC*
Field name	*EXAMPLE_CHAR*

6. Create a new service called *PSFT81_SERVICE.*
7. Create a new service operation.
   a. Add a service operation of type asynchronous-one way to the *PSFT81_SERVICE* and name it *PSFT81_SERVICE_OPR.*
   b. Add *EXAMPLE_PSFT_MSG* as the message.
   c. Add *EXAMPLE_QUEUE* as the queue.
   d. Configure the service operation security for this service operation.
8. Add an inbound routing for the *PSFT81_SERVICE_OPR* service operation with the source node being the 8.1 system and the destination being the 8.55 system.
9. Add an outbound routing for the *PSFT81_SERVICE_OPR* service operation with the source node being the 8.55 system and the destination being 8.1 system.

10. Open the *EXAMPLE_WORKREC* record. Add the following PeopleCode to the FieldChange event for the *TEST* field:

```
&message = CreateMessage(Operation.PSFT81_SERVICE_OPR);

/* get the buffer data */
&rowset = GetLevel0();

/* copy buffer data to the message */
&message.CopyRowset(&rowset);

/* send the message */
&message.Publish();
```

11. Go to the connector information for the new node. Set the **Connector ID** to *PSFT81TARGET*. Set the URL property to the address of the gateway servlet on the PeopleSoft 8.1 system. For example:

```
http://<theServerNameAndPort>/servlets/gateway
```

Setting Up Data on the PeopleSoft 8.1 System

To set up data on the PeopleSoft 8.1 system:

1. In PeopleSoft Application Designer, create a new field called *EXAMPLE_CHAR*. This should be a mixed-case character field of size *20*.
2. Create a new record.
    a. Name the record *EXAMPLE_REC*.
    b. Add the *EXAMPLE_CHAR* field to this record, set it as the key, and save the definition.
    c. Build the physical table for this record.
3. Create a new message channel called *EXAMPLE_CHANNEL*. On the properties dialog box, set the **Status** to **Run**. Configure the security for the message monitor so that the channel can be displayed.
4. Create a new message.
    a. Open the properties and select the **Active** box for the **Status**.
    b. Set the **Message Channel** to **EXAMPLE_CHANNEL**.
    c. Add the *EXAMPLE_REC* record to *VERSION_1* of this message.
    d. Save the message as *EXAMPLE_PSFT_MSG*.
5. Add the subscription *ExampleSubscription* to the *EXAMPLE_PSFT_MSG*. Use the following PeopleCode in the subscription body:

```
/* get the incoming message */
&msg = GetMessage();
&msgXML = &msg.GenXMLString();

/* and write it to a file */
&file = GetFile("PSFT81msg.txt", "w", "UTF8");
&file.writeString(&msgXML);
&file.close();
```


6. Create a new message node, using the name of the PeopleSoft 8.55 node. Add a **Location** to this node with the following format:

   ```
   http://<serverName:port>/PSIGW/PS81ListeningConnector
   ```

   The server name and port you specify must correspond to the integration gateway address of the PeopleSoft 8.55 system.

7. Open the *EXAMPLE_CHANNEL*. Add a new routing rule to the channel, where the direction is **Both** and the message node name is that of the PeopleSoft 8.55 node.

8. In the Message Monitor, invoke the Gateway Administration servlet and add the PeopleSoft 8.1 node to the PeopleSoft handler.

9. Open the Message Monitor and verify that the *EXAMPLE_CHANNEL* is running.

Using the PeopleSoft 8.1 Target Connector

In the example presented in this section, you will use the PeopleSoft 8.1 target connector to send a message from a PeopleSoft 8.55 system to a PeopleSoft 8.1 system.

To send a message from a PeopleSoft 8.55 system to a PeopleSoft 8.1 system:

1. In the PeopleSoft Pure Internet Architecture on the PeopleSoft 8.55 system, open the *EXAMPLE_PAGE*. Enter text into the edit box, and press the *Test* button. Wait for a minute or two to allow the systems to process the message.

2. On the PeopleSoft 8.55 system, open the Service Operations Monitor to view the details of the outbound message.

3. On the PeopleSoft 8.1 system, open up the Message Monitor to view the details of the received message. Open the PSFT81msg.txt file created by the subscription PeopleCode to see the body of the message.

Using the PeopleSoft 8.1 Listening Connector

In the example presented in this section, you will use the PeopleSoft 8.1 listening connector to send a message from a PeopleSoft 8.1 system to a PeopleSoft 8.55 system.

To send a message from a PeopleSoft 8.1 system to a PeopleSoft 8.55 system:

1. On the PeopleSoft 8.1 system, open PeopleSoft Application Designer and open the *EXAMPLE_PSFT_MSG* message. Right-click *VERSION_1* and select **Create test message**. The Create Test Message window appears.

2. Expand **Transaction** in the tester window. Set the value for *EXAMPLE_CHAR*. Open the PSCAMA section and set the AUDIT_ACTN to *A* and click **OK**. A message is published. Wait a minute or two before proceeding to allow the message to be processed by both nodes.

3. On the PeopleSoft 8.1 system, open the Message Monitor to view the details of the outbound message.

4. On the PeopleSoft 8.55 system, open the Service Operations Monitor to view the details of the received message.

Example 4: Using the JMS Connectors

This section discusses using the JMS listening and JMS target connectors.

Understanding the JMS Connector Examples

The examples in this section are intended to be generic and independent of the JMS provider being used. Because of this, in certain steps general instructions are provided. The actual details of the task will depend on the provider being used – and may be rather involved. Please refer to the appropriate documentation.

The error queue is not configured in the examples. However, configuring the error queue may be desirable should issues arise while trying the examples.

The examples in this section focus on queues, but the process for using the JMS connectors when working with topics is essentially the same.

Related Links

Working With the JMS Connectors

Prerequisites

To use the examples in this section, a JMS provider must be configured and running. Please refer to the provider's documentation for instructions on how to accomplish these tasks. Ensure that messages can be sent to topics and queues before proceeding with the examples.

For the JMS target connector example, you will need a utility to consume and view the messages created. For the JMS listening connector example, you will need a utility to create the messages. The exact details of these utilities depend on the provider. Some may provide an administrative console that you can use to view the contents of topics and queues, and possibly send new messages to them. Other providers may include sample Java programs that you can use to interact with the provider. Refer to the provider's documentation for further details.

A special case exists for testing the JMS listening connector and queues when the provider is IBM MQSeries. In this instance, use Send Master to test the JMS listening connector.

Related Links

"Using JMS Projects" (Integration Broker Testing Utilities and Tools)

Using the JMS Target Connector

In this example, PeopleSoft Integration Broker will generate a JMS message, which will be consumed outside of the PeopleSoft system.

To use the JMS target connector:

1. On the JMS provider, create a JMS Connection Factory with the JNDI name *ExampleConnectionFactory*.

2. On the JMS provider, create a JMS Queue with the JNDI name *ExampleQueue*.

3. In PeopleSoft Application Designer, open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the TEST field:

```
/* create an XML document */
&xmldata = "<?xml version='1.0'?><ConnectorTest/>";
&xmlDoc = CreateXmlDoc(&xmldata);
&rootNode = &xmlDoc.documentelement;

/* add text to the document */
&descNode = &rootNode.AddElement("TestNode");
&descNode.NodeValue = "Sending a message to a JMS queue.";

/* and send it out in an async request */

&MSG = CreateMessage(Operation.EXAMPLE_SERVICE_ASYNC_OPR);
&MSG.SetXmlDoc(&xmlDoc);
%IntBroker.Publish(&MSG);

MessageBox(0, "", 0, 0, "Message sent.");
```

4. In the PeopleSoft Pure Internet Architecture, open the node definition for *TARGETNODE*. Set the **Connector ID** to *JMSTARGET*. Set the values for the following properties:

Property	***Value***
JMSFactory	*ExampleConnectionFactory.*
JMSProvider	Name of the provider being used.
JMSUrl	Connection URL for the provider.
JMSQueue	*ExampleQueue.*
JMSUserName	The username on the JMS provider.
JMSPassword	The encrypted password for the user ID.

5. Test the connector:

   a. Open the test page, and click on the **Test** button.

   b. Verify that the message was sent to the queue. The exact mechanism for doing depends on the provider or utility that you are using.

Using the JMS Listening Connector

In this example, you will use the JMS listening connector to send a message to the JMS provider. PeopleSoft Integration Broker will consume the message.

To use the JMS listening connector:

1. On the JMS provider, create a JMS Connection Factory with the JNDI name *ExampleConnectionFactory*.
2. On the JMS provider, create a JMS Queue with the JNDI name *ExampleQueue*.
3. In PeopleSoft Application Designer, create a application package and application class. In the application class, put the following PeopleCode in the OnRequest function:

```
Local XmlDoc &xmldoc;

&xmldoc = &_MSG.GetXmlDoc(); /*&_msg is the parameter*/
/* and write it to a file */
Local File &theFile = GetFile("JMSRequest.txt", "W", "UTF8");

&theFile.WriteString(&xmldoc.GenXmlString());
&theFile.Close(); /* create the reponse message */
Local Message &outmsg;
&outmsg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR,
%IntBroker_Response);
/* build the body of the response */

Local string &xmldata = "<?xml version='1.0'?><ConnectorTest/>";
&xmldoc = CreateXmlDoc(&xmldata);

Local XmlNode &rootNode = &xmldoc.DocumentElement;

Local XmlNode &descNode = &rootNode.AddElement("ResponseMessage");
&descNode.NodeValue = "This wasgenerated in the OnRequest handler.";

/* add the body to the message */
&outmsg.SetXmlDoc(&xmldoc);

/* send the response message */
Return &outmsg;
```

4. In the PeopleSoft Pure Internet Architecture, open the handler tab on the service operation *EXAMPLE_SERVICE_OPR,* and set the application class package, class and method name as you defined above.
5. In the integrationGateway.properties file, uncomment the following line:

```
ig.jms.Queues=1
```

6. Set the following properties to the values indicated:

Property	***Value***
ig.jms.Queue1	*ExampleQueue*
ig.jms.Queue1.Provider	*<the name of the provider>*
ig.jms.Queue1.JMSFactory	*ExampleConnectionFactory*
ig.jms.Queue1.Url	*<connection URL for the provider>*
ig.jms.Queue1.Use	*< the userid >*

Property	*Value*
ig.jms.Queue1.Password	*<the encrypted password for the userid. >*
ig.jms.Queue1.MessageName	*EXAMPLE_SERVICE_OPR.VERSION_1*
ig.jms.Queue1.RequestingNode	*SOURCENODE*
ig.jms.Queue1.DestinationNode	*<the name of the local node >*

7. Deploy and start the JMSListeningConnectorAdministrator servlet.

   See Using the JMS Listening Connector.

8. Test the connector:

   a. Send a text message to the example JMS queue. Set the text of the message to:

```
<?xml version="1.0"?>
<ConnectorTest>
     <TestNode>Sending a message to the JMS Listening Connector.</TestNod⇒

e>
</ConnectorTest>
```

   b. Check the message logs and the file named in the OnRequest method of application class . The message should be present in both.

Example 5: Using the AS2 Connectors

This section discusses using the AS2 listening and AS2 target connectors.

Note: AS2 Connectors will be desupported in future PeopleTools release.

Understanding the AS2 Connector Examples

The purpose of the AS2 protocol is to allow the secure exchange of EDI data over the internet with trading partners. In the simplest case of an AS2 Message exchange, a sender packages data into an AS2 message structure and sends the message to trading partner over HTTP. Any kind of data can be transferred using AS2, including XML, EDI, text and binary.

The examples in this section demonstrate using the AS2 target connector to send an XML message to an external trading partner and using the AS2 listening connector to receive an XML message from a trading partner.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Related Links

Working With the AS2 Connectors

Prerequisites

To use the examples in this section, security certificates must be setup and registered in the keystore on the source and target machines. Take note of the certificate alias name for both the source or signer and the target or recipient servers, as you will need this information to set connector properties.

Verify that messages can be sent to and received from the AS2 external trading partner over HTTP before proceeding with the examples.

For the AS2 target connector example, you will need a third-party application to consume and view the messages created. For the AS2 listening connector example, you will need a third-party application to create and deliver the messages.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

Related Links

Understanding Securing Integration Environments

Using the AS2 Target Connector

In this example PeopleSoft Integration Broker will generate an AS2 message and send it to a trading partner using HTTP. The external trading partner consumes the message. This example shows the tasks to perform to receive an MDN response message back synchronously or asynchronous.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

To use the AS2 target connector:

1. In PeopleSoft Application Designer open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the **TEST** field:

```
/*create an XML document */
Local string &xmldata;
Local XmlDoc &xmlDoc;
Local XmlNode &rootNode, &descNode;
Local boolean &result;

&xmldata = "<AS2ConnectorTest/>";

&xmlDoc = CreateXmlDoc("");
&rootNode = &xmlDoc.CreateDocumentElement("AS2ConnectorTest");
&rootNode = &xmlDoc.DocumentElement;

/* add text to the document */
&descNode = &rootNode.AddElement("TestNode");
&descNode.NodeValue = "Sending a AS2 message.";

&MSG = CreateMessage(Operation.EXAMPLE_SERVICE_ASYNC_OPR);
&MSG.SetXmlDoc(&xmlDoc);

/* and send it out in an async request */
```



```
%IntBroker.Publish(&MSG);

MessageBox(0, "", 0, 0, "AS2 Message sent.");
```

2. In the PeopleSoft Pure Internet Architecture, open the node definition for TARGETNODE. Set the Connector ID to *AS2TARGET*. Set the values for the following required properties:

Property Name	***Value***
URL	Specify the URL to which messages are sent using this connector.
AS2To	Specify the name of the sending node.
AS2From	Specify the name of the receiving node.
RecipientCertAlias	Specify the alias of the receiving certificate.
SignersCertificateAlias	Specify the alias of the signing certificate.

3. Add an outbound asynchronous transaction on the *AS2TARGETNODE*, to identify that the message *EXAMPLE_MESSAGE, VERSION_1* will be sent to the URL location.

4. Set the following properties in the integrationGateway.properties file. Use PSCipher.bat utility located at <PIA_HOME>\webserv\peoplesoft to encrypt the keystore password.

```
#AS2 Log Directory, logs all incoming and outgoing AS2 request and responses.
#Uncomment and specify the correct directory name to enable logging.
ig.AS2.LogDirectory = c://temp//as2

#AS2 Properties
#Uncomment the following two lines to specify your keystore and AS2 properties
ig.AS2.KeyStorePath=KeyStore Location (use // for windows path)
ig.AS2.KeyStorePassword=EncryptedKeyStorePassword
ig.AS2.AS2Directory=Location of AS2 Certificates  (required for Async MDN Type⇒

)
ig.AS2.LogDirectory=Path to store AS2 Log Files (optional)

Examples
ig.AS2.KeyStorePath=C://pt846-112-R2//webserv//peoplesoft//keystore//pskey
ig.AS2.KeyStorePassword=*** Encrypted password ***
ig.AS2.AS2Directory=c://temp//as2
ig.AS2.LogDirectory = c://temp//as2//logs
```

5. If the MDN response is synchronous, go to step 8.

   If the MDN response is asynchronous, verify the delivered node named *ASYNC_MDN* exists.

6. Verify that the node *ASYNC_MDN* has an active incoming asynchronous routing for the service operation *ASYNC_MDN_RESPONSE. VERSION_1*.

7. Verify that the delivered queue *AS2_CHANNEL* is not in *Pause* mode.

8. Test the connector.

   a. Open the test page, and click on the **Test** button.

b. Verify that the message was sent to the recipient. The exact mechanism for doing so depends on the AS2 trading partner you are using.

c. Verify that the MDN response was sent back to the source. The exact mechanism for doing so depends on the AS2 trading partner you are using.

9. If the MDN type is set to *Async,* verify that the A*SYNC_MDN_RESPONSE* was received.

Using the AS2 Listening Connector

In this example, you will use the AS2 listening connector to receive a message sent by the AS2 trading partner, and return an MDN synchronous or asynchronous response. Perform all tasks on the target machine. PeopleSoft Integration Broker will consume the message.

Note: PeopleSoft recommends using the Oracle SOA Suite B2B component for all EDI integrations, including those based on the AS2 specification.

To use the AS2 listening connector:

1. In the PeopleSoft Pure Internet Architecture, choose the node that corresponds to the AS2 trading partner sending the message.

2. Insert an inbound asynchronous routing corresponding to the service operation *EXAMPLE_REQUEST_ASYNC_OPR.VERSION_1* expected.

3. Insert an outbound asynchronous routing corresponding to the service operation *EXAMPLE_RESPONSE_ASYNC_OPR.VERSION_1* as a reply.

4. In PeopleSoft Application Designer, create an application package and application class, and provide a method OnNotify with the following PeopleCode:

```
Local XmlDoc &xmlDoc;
   Local File &theFile;
   Local Message &msg;
  Local XmlDoc &MsgXmlDoc,  &xmlDoc;
   Local XmlNode &rootNode, &descNode;

   /* get the body of the incoming message */
   &MsgXmlDoc = &MSG.GetXmlDoc(); /* and write it to a file */
   &theFile = GetFile("AS2Request.txt", "W", "UTF8");
   &theFile.WriteString(&MsgXmlDoc.GenXmlString());
   &theFile.Close();

   &xmlDoc = CreateXmlDoc("");

   &rootNode = &xmlDoc.CreateDocumentElement("ConnectorTest");
   &rootNode = &xmlDoc.DocumentElement; /* add text to the document */

   &descNode = &rootNode.AddElement("ResponseMessage");
   &descNode.NodeValue = "This was generated in the OnRequest event.";
   /* send the response message */

   &msg = CreateMessage(Operation.EXAMPLE_RESPONSE_ASYNC_OPR);

   &msg.SetXmlDoc(&xmlDoc);

   /* and send it out in an async request */
   %IntBroker.Publish(&msg);
```


5. In the PeopleSoft Pure Internet Architecture, open the handler tab on the service operation EXAMPLE_RESPONSE_ASYNC_OPR, and set the application package, class name and method.

6. In the integrationGateway.properties file, set the following properties to the values indicated:

```
#AS2 Properties
#Uncomment the following two lines to specify your keystore and AS2 properties
ig.AS2.KeyStorePath=KeyStore Location (use // for windows path)
ig.AS2.KeyStorePassword=EncryptedKeyStorePassword
ig.AS2.LogDirectory=Path to store AS2 Log Files (optional)

#example:
ig.AS2.KeyStorePath=C://pt846-112-R2//webserv//peoplesoft//keystore//pskey
ig.AS2.KeyStorePassword=*** Encrypted password ***
ig.AS2.LogDirectory = c://temp//as2//logs
```

In the following required properties, replace the <*SOURCENODE*> with the name of the AS2 trading partner source node, and <*TARGETNODE*> with the name of the local target node. Continue to set the value of the property.

```
# CertificateAlias is the certificate of AS2 Listening Node.
# SignerCertificateAlias is the certificate of AS2 trading partner of Listenin⇒

g Node.
ig.AS2.QE_<SOURCE>.<TARGET>.CertificateAlias= Target Machine Alias
ig.AS2. <SOURCE>.<TARGET>.SignerCertificateAlias=Source Machine Alias

#example:
ig.AS2.PSFT_SRC_NODE.PSFT_TGT_NODE.CertificateAlias=<GeneratedAS2certificateal⇒

ias>
ig.AS2.PSFT_SRC_NODE.PSFT_TGT_NODE.SignerCertificateAlias=<GeneratedAS2certifi⇒

catealias>
```

The following values only need to be set if the incoming data does not contain the appropriate AS2To and AS2From values in the header of the message. It is best to leave these values in the request message header and leave these properties commented out.

```
#This map translate AS2From and AS2To to a different node name.
#This property is not required if you would use AS2FROM and AS2TO http header.
ig.AS2.AS2ListenerMap.From.<SOURCEALIAS> = Specify the Source Node Name
ig.AS2.AS2ListenerMap.To.<TARGETALIAS> = Specify the Target Node Name

#example:
ig.AS2.AS2ListenerMap.From.QE_SOURCE= PT_LOCAL
ig.AS2.AS2ListenerMap.To. QE_IBTGT= AS2TARGETNODE
```

7. Test the connector:

   a. Send a text message to the example AS2 queue. Name the message *EXAMPLE_REQUEST_MSG*.

   b. Set the text of the message to:

```
<?xml version="1.0"?>
<ConnectorTest>
<TestNode>Sending a message to the AS2 Listening Connector.</TestNode>
</ConnectorTest>
```

   c. Check the file named in the subscription PeopleCode. The default location for this file is *<PS_CFG_HOME>\appserv\<DOMAIN_NAME>\Files.* The message contents should be present.

d. If the MDN type is asynchronous, verify that the AS2 trading partner received the *ASYNC_MDN_RESPONSE*.

Example 6: Using the FTP Target Connector

This sections discusses how to use the FTP target connector to:

- Upload files to an FTP server.
- Download files from an FTP server.

Note: FTP Target Connector will be desupported in future PeopleTools release.

Prerequisites

For the examples presented in this section, you must have an active FTP server, as well as an account on that server.

Uploading Files to FTP Servers

To upload a file to an FTP server:

1. In PeopleSoft Application Designer, open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the **TEST** field:

```
&msg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);

&xmldata = "<?xml version='1.0'?><ConnectorTest/>";

/* create an XmlDoc */
&xmlDoc = CreateXmlDoc(&xmldata);
&rootNode = &xmlDoc.documentelement;
&descNode = &rootNode.addelement("FTPtest");
&descNode.nodevalue = "This text will be uploaded";

/* put the XML in the message */
&msg.setxmldoc(&xmlDoc);

/* send the request */
&response = %IntBroker.SyncRequest(&msg);
```

2. In the PeopleSoft Pure Internet Architecture, open the *TARGETNODE* node definition.
   a. On the Node Definitions-Connectors tab, set the **Connector ID** to *FTPTARGET*.
   b. Set the following properties to the values indicated:

Property	***Value***
HOSTNAME	Specify the IP address or name of the FTP server for the connection.

Property	*Value*
METHOD	*PUT*
USERNAME	Enter the FTP server login ID.
PASSWORD	Enter the password for the login to the FTP server. This password must be encrypted. Use the Password Encryption Utility at the bottom of the page to encrypt the password, if necessary

3. In the PeopleSoft Pure Internet Architecture, open the EXAMPLE_PAGE page and click the **Test** button.

   Login to the FTP server and check for the file. Open the file and verify the contents.

Downloading Files From FTP Servers

To download a file from an FTP server:

1. Create an XML file with the following contents and place the file on an FTP server.

```
<?xml version="1.0"?>
<ConnectorTest>
<TestNode>This message will be downloaded from an FTP server.</TestNode>
</ConnectorTest>
```

2. In PeopleSoft Application Designer, open the EXAMPLE_WORKREC record and add the following PeopleCode to the FieldChange event for the **TEST** field:

```
&msg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);

&xmldata = "<?xml version='1.0'?><ConnectorTest/>";

/* create an XmlDoc */
&xmlDoc = CreateXmlDoc(&xmldata);

/* put the XML in the message */
&msg.setxmldoc(&xmlDoc);

/* send the request */
&response = %IntBroker.SyncRequest(&msg);

/* display the contents */
&xmlDoc = &response.getxmldoc();
MessageBox(0, "", 0, 0, &xmlDoc.genxmlstring());
```

3. In the PeopleSoft Pure Internet Architecture, open the *TARGETNODE* node definition.

   a. On the Node Definitions-Connectors tab, set the **Connector ID** to *FTPTARGET*.

   b. Set the following properties to the values indicated:

Property	*Value*
HOSTNAME	Specify the IP address or name of the FTP server for the connection.
METHOD	*GET*
FILENAME	Specify the name of the file.
USERNAME	Enter the FTP server login ID.
PASSWORD	Enter the password for the login to the FTP server. This password must be encrypted. Use the Password Encryption Utility at the bottom of the page to encrypt the password, if necessary

4. In the PeopleSoft Pure Internet Architecture, open the EXAMPLE_PAGE page and click the **Test** button.

   The contents of the XML file will display in the message box.

Example 7: Using the SFTP Target Connector

This section discusses how to:

- Upload files to an SFTP server.
- Download binary files from an SFTP server.

Prerequisites

For the examples presented in this section, you must have an active SFTP server, as well as an account on that server.

Uploading Files to an SFTP Server

To upload a file to an SFTP server:

1. Add a new URL object.
   a. Select **PeopleTools** > **Utilities** > **Administration** > **Maintain URLs.**

      The URL Maintenance page appears.

   b. Add the following values:

Field	*Value*
URL Identifier	*SFTPTESTURL*
Description	*SFTP Server URL*
URLID	Enter the appropriate URL. For example: `sftp://<userid>:<password>@<address>:<port>⇒`

c. Save the changes.

2. In PeopleSoft Application Designer, open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the TEST field:

```
Local array of Message &messages;
Local array of Message &responses;
Local Message &MSG;
Local array of string &nodenames;
Local string &node;

&xmldata = "<?xml version='1.0'?><sftpTest/>";

&xmlDoc = CreateXmlDoc(&xmldata);

&rootNode = &xmlDoc.documentelement;
&descNode = &rootNode.AddElement("Payload");
&descNode.NodeValue = "Generated at " | %Datetime;


&messages = CreateArrayRept(&MSG, 0);
&nodenames = CreateArrayRept(&node, 0);

&nodenames [1] = "TARGETNODE";
&messages [1] = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);
&messages [1].SetXmlDoc(&xmlDoc);

&responses = %IntBroker.SyncRequest(&messages, &nodenames);
```

3. In the PeopleSoft Pure Internet Architecture, open the TARGETNODE node definition and do the following:

a. On the Node Definitions-Connectors tab, set the Connector ID to *SFTPTARGET.*

b. Set the following properties to the values indicated:

Property	*Value*
BASE64ENCODE	*Y.*
CHARSET	*UTF-8.*
METHOD	*PUT.*

Property	*Value*
REMOTEFILENAME	*SFTPTestFile.txt.*
URL	*SFTPTESTURL.*

4. In the PeopleSoft Pure Internet Architecture, open the EXAMPLE_PAGE page and click the **Test** button.

   Login to the SFTP server and check for the file. Open the *SFTPTestFile.txt* file and verify the contents.

Downloading Binary Files from SFTP Servers

To download a binary file from an SFTP server:

1. Put a binary file in an accessible location on the SFTP server.

   For this example, the file is called *image.jpg.*

2. Add a new URL object.

   a. Select **PeopleTools** > **Utilities** > **Administration** > **Maintain URLs.**

      The URL Maintenance page appears.

   b. Add the following values:

Field	*Value*
URL Identifier	*SFTPTESTURL*
Description	*SFTP Server URL*
URLID	Enter the appropriate URL. For example: `sftp://<userid>:<password>@<address>:<port>⇒`

   c. Save the changes.

3. In PeopleSoft Application Designer, open the *EXAMPLE_WORKREC* record and add the following PeopleCode to the FieldChange event for the TEST field:

```
Local array of Message &messages;
Local array of Message &responses;
Local Message &MSG;
Local array of string &nodenames;
Local string &node;
```



```
&xmldata = "<?xml version='1.0'?><sftpTest/>";

&xmlDoc = CreateXmlDoc(&xmldata);

&messages = CreateArrayRept(&MSG, 0);
&nodenames = CreateArrayRept(&node, 0);

&nodenames [1] = "TARGETNODE";
&messages [1] = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);
&messages [1].SetXmlDoc(&xmlDoc);

&responses = %IntBroker.SyncRequest(&messages, &nodenames);


&theTempFile = GetFile("C:\work\aNewImage.jpg", "W", %FilePath_Absolute);
If &theTempFile.IsOpen Then
   &theTempFile.writeBase64StringtoBinary(&responses [1].GetContentString());
   &theTempFile.Close();
End-If;
```

4. In the PeopleSoft Pure Internet Architecture, open the TARGETNODE node definition and do the following:

   a. On the Node Definitions-Connectors tab, set the Connector ID to *SFTPTARGET.*

   b. Set the following properties to the values indicated:

Property	***Value***
BASE64ENCODE	*N.*
CHARSET	*UTF-8.*
METHOD	*GET.*
REMOTEFILENAME	*image.jpg*
URL	*SFTPTESTURL.*

5. In the PeopleSoft Pure Internet Architecture, open the EXAMPLE_PAGE page and click the Test button.

   Check the local directory for the file. Open the file and verify the contents.

Example 8: Using the SMTP Target Connector

This section provides an example of how to use the Simple Mail Transfer Protocol (SMTP) target connector to send an email message using an SMTP server.

Prerequisites

For this example, you must have an active SMTP server as well as an active email account to receive the message.

Sending Email Messages to SMTP Servers

To send an email message to an SMTP server using the SMTP target connector:

1. In PeopleSoft Application Designer, open the EXAMPLE_WORKREC record and add the following PeopleCode to the FieldChange event for the **TEST** field:

```
&msg = CreateMessage(Operation.EXAMPLE_SERVICE_OPR);

&xmldata = "<?xml version='1.0'?><ConnectorTest/>";

/* create an XmlDoc */
&xmlDoc = CreateXmlDoc(&xmldata);
&rootNode = &xmlDoc.documentelement;
&descNode = &rootNode.addelement("SMTPtest");
&descNode.nodevalue = "This xml will appear in the email";

/* put the XML in the message */
&msg.setxmldoc(&xmlDoc);

/* send the request */
&response = %IntBroker.SyncRequest(&msg);
```

2. In the PeopleSoft Pure Internet Architecture, open the *TARGETNODE* node definition.

   a. On the Node Definitions-Connectors tab, set the **Connector ID** to *SMTPTARGET*.

   b. Set the following properties to the values indicated:

Property	***Value***
DestEmailAddress	Set this property to the email address to which the email will be sent.
SourceEmailAddress	Set this property to the email address from which you are sending the message.

3. Access the integrationGateway.properties file. Locate the following line and replace <mailServerName> with the name of the SMTP server.

```
ig.connector.smtptargetconnector.host=<mailServerName>
```

4. In the PeopleSoft Pure Internet Architecture, open the EXAMPLE_PAGE page and click the **Test** button to send the message.

   Check the destination email account for the message. Since the message is being passed through one or more SMTP servers, there may be some propagation delay and the message might not be received immediately.

Chapter 17

Using the Integration Broker Connector SDK

Understanding the PeopleSoft Integration Broker Connector SDK

This section discusses:

- The PeopleSoft Integration Broker Connector SDK.
- SDK contents.
- SDK location.
- SDK connector example.

The PeopleSoft Integration Broker Connector SDK

Target connectors generate message requests, send them to integration participants, wait for responses from participants, and deliver the responses back to the gateway manager. Listening connectors receive message requests from integration participants, send them to the gateway manager, and deliver responses back to the integration participants.

PeopleSoft Integration Broker is bundled with connectors for use with PeopleSoft, HTTP, Java Messaging Service (JMS), PeopleSoft 8.1x, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP) communication formats. You can use the PeopleSoft Integration Broker Connector SDK to build and implement connectors for other communication formats and application requirements.

SDK Contents

The PeopleSoft Integration Broker Connector SDK includes:

- Java classes that are required for creating connectors (including IBResponse and IBRequest objects).
- Sample code for listening and target connector classes.
- A Send Master utility to test connectors.
- A Simple Post utility that enables third-party systems to post messages to the integration gateway.

SDK Location

The following table lists the location of the SDK and its contents.

Item	*Location*
SDK	<PIA_HOME>\webserv\<DOMAIN>\applications⇒ \ peoplesoft\PSIGW.war\WEB_INF\SDK
Java classes	<PIA_HOME>\webserv\<DOMAIN>\applications⇒ \peoplesoft\ PSIGW.war\WEB-INF\classes
Sample code for listening and target connector classes	<PIA_HOME>\webserv\<DOMAIN>\applications⇒ \peoplesoft\ PSIGW.war\WEB_INF\SDK\src
Send Master utility	Microsoft Windows: <PIA_HOME>\webserv\<DOMAIN>\piabin\Start⇒ SendMaster.bat UNIX: <PIA_HOME>\webserv\<DOMAIN>\piabin\Star⇒ tSendMaster.sh
Simple Post utility	<PIA_HOME>\webserv\<DOMAIN>\applications⇒ \peoplesoft\ PSIGW.war\WEB-INF\classes\com\peoplesoft⇒ \pt\simplepost

SDK Connector Examples

Four sample connectors are provided as part of the SDK:

- ExampleListeningConnector.java
- ExampleServletListeningConnector.java
- ExampleTargetConnector.java
- SimpleFileTargetConnector.java

These connectors can be used as the basis for new development.

The ExampleListeningConnector and the ExampleServletListeningConnector highlight the differences between a minimalist listening connector and one intended to be run as a servlet.

The ExampleTargetConnector shows the basic requirements of a working target connector.

The SimpleFileTargetConnector is an example of a target connector written to perform a specific task: write outgoing message data to the file system.

To compile these connectors you must set the Java classpath to include the Integration Broker classes in:

```
<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEBINF\classes
```

and

```
<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEBINF\lib\mail.jar
```

The Java class path must also be set to include the runtime Jar file for the installed web server, for example:

```
weblogic.jar for a WebLogic installation
```

Understanding Connector Development and Implementation

This section discusses connector development and implementation.

Understanding Developing Connectors

You can produce new connectors in different ways, based on whether you want to create a listening connector or a target connector.

Listening connectors use standard connector interface and gateway services to link to the integration gateway. Although a Java interface object is not used for listening connectors, the listening connectors still must adhere to a standard scheme of logic to drive requests to, and to process responses from, the integration gateway.

Target connectors must implement a Java interface to become valid target connectors in the integration gateway. This ensures a standard interface for the gateway manager so that it can manage each target connector in a streamlined way.

To develop connectors, you:

1. Develop a connector class.
2. Install the connector class.
3. Register the connector.

Understanding General Connector Class Development Considerations

While implementations vary greatly, when you develop connector classes, you should incorporate specific functionality.

Input and Output Formats That Are Exchanged Through Connectors

For a target connector to handle input and output formats that are exchanged with its intended recipient, it must transform the PeopleSoft Integration Broker request (IBRequest) into a message that is formatted for the intended external system.

For instance, the HTTP target connector that is delivered with PeopleSoft Integration Broker gathers HTTP headers and cookies from the IBRequest and formulates the appropriate HTTP message, complete with the actual message content, so that it can be delivered to its destination. When the response comes back, the connector creates a PeopleSoft Integration Broker response (IBResponse) by using the response string.

For a listening connector to handle input and output formats that are exchanged with its requestor, it must transform the incoming message into an IBRequest. For example, the HTTP listening connector that is delivered with PeopleSoft Integration Broker recognizes SOAP messages and retrieves query string arguments, HTTP headers, and cookies. It then formats all of this information to create the IBRequest so that PeopleSoft Integration Broker can converse with it. When the response comes back, the HTTP listening connector reads the IBResponse and sends its output message content back to the requesting system.

Interaction Between Local and External Systems

A target connector interacts with an external system by sending it information and by retrieving the response.

For example, to accomplish this interaction, the HTTP target connector that is delivered with PeopleSoft Integration Broker uses various HTTP-specific classes to send messages through HTTP and to handle the external system being down, security (through HTTPS), and so forth.

A listening connector interacts with an external system by receiving requests from the external system and returning responses that the external system understands. For example, to accomplish this interaction, the HTTP listening connector that is delivered with PeopleSoft Integration Broker uses a servlet to receive and reply to incoming HTTP messages.

Developing Target Connector Classes

This section discusses target connector class development and discusses how to:

- Use the target connector interface.
- Build introspection into target connectors.
- Build error handling and logging into target connectors.

Using the Target Connector Interface

As with PeopleSoft-provided target connectors, the integration gateway dynamically invokes custom target connectors through the gateway manager. Target connectors must adhere to a standard structure as defined in the target connector interface.

```
public interface TargetConnector {
   IBResponse send(IBRequest request) throws
```



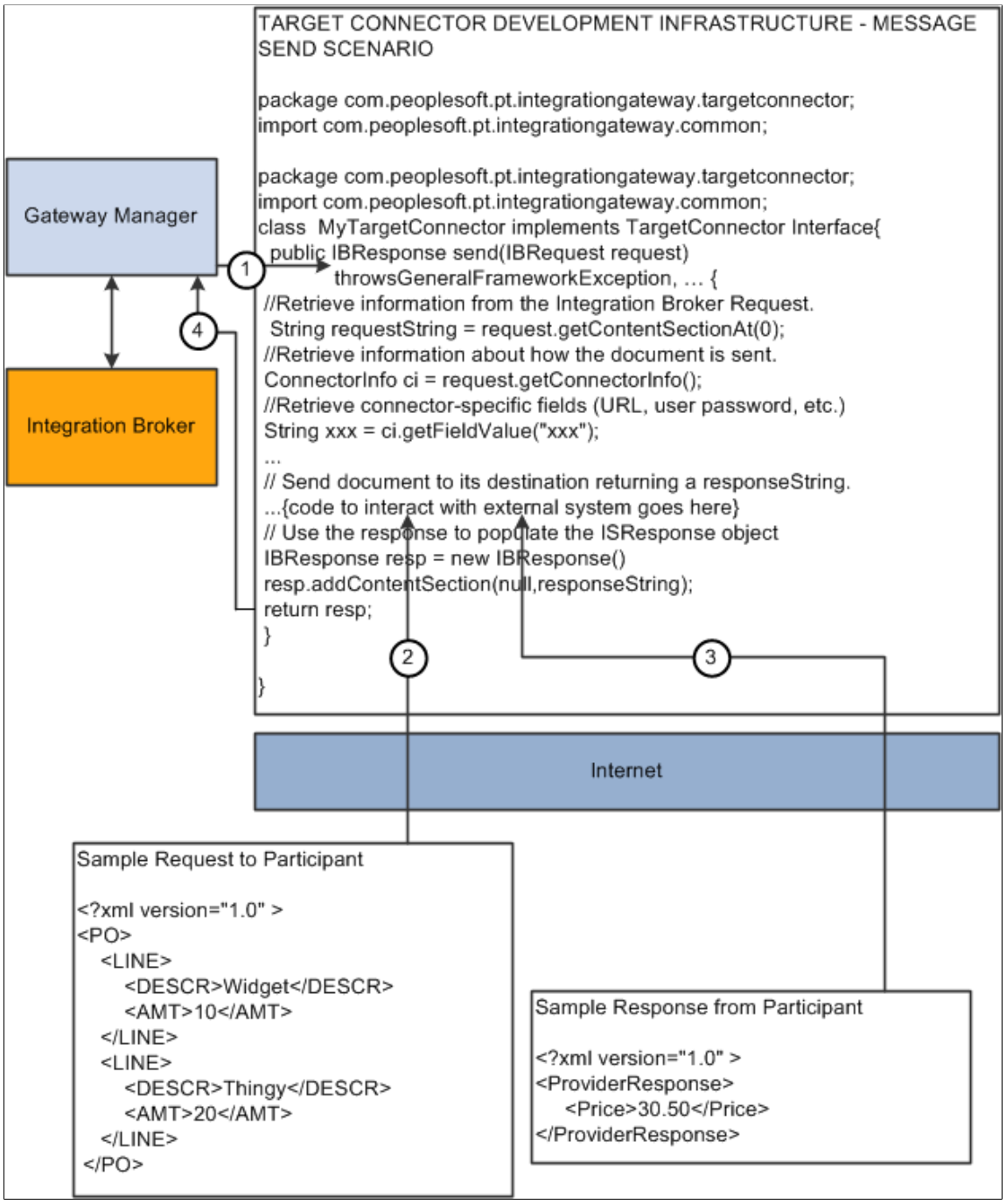
```
                        GeneralFrameworkException,
                        DuplicateMessageException,
                        InvalidMessageException,
                        ExternalSystemContactException,
                        ExternalApplicationException,
                        MessageMarshallingException,
                        MessageUnmarshallingException;

    IBResponse ping(IBRequest request) throws
                        GeneralFrameworkException,
                        DuplicateMessageException,
                        InvalidMessageException,
                        ExternalSystemContactException,
                        ExternalApplicationException,
                        MessageMarshallingException,
                        MessageUnmarshallingException;

ConnectorDataCollection introspectConnector();
```

Use the Send method to send a request to an external system and to retrieve its response. The gateway manager passes the request to this method and expects a response to be returned.

The Ping method enables PeopleSoft Integration Broker to verify the availability of a site. The Integration Broker Monitor can also invoke the Ping method explicitly.

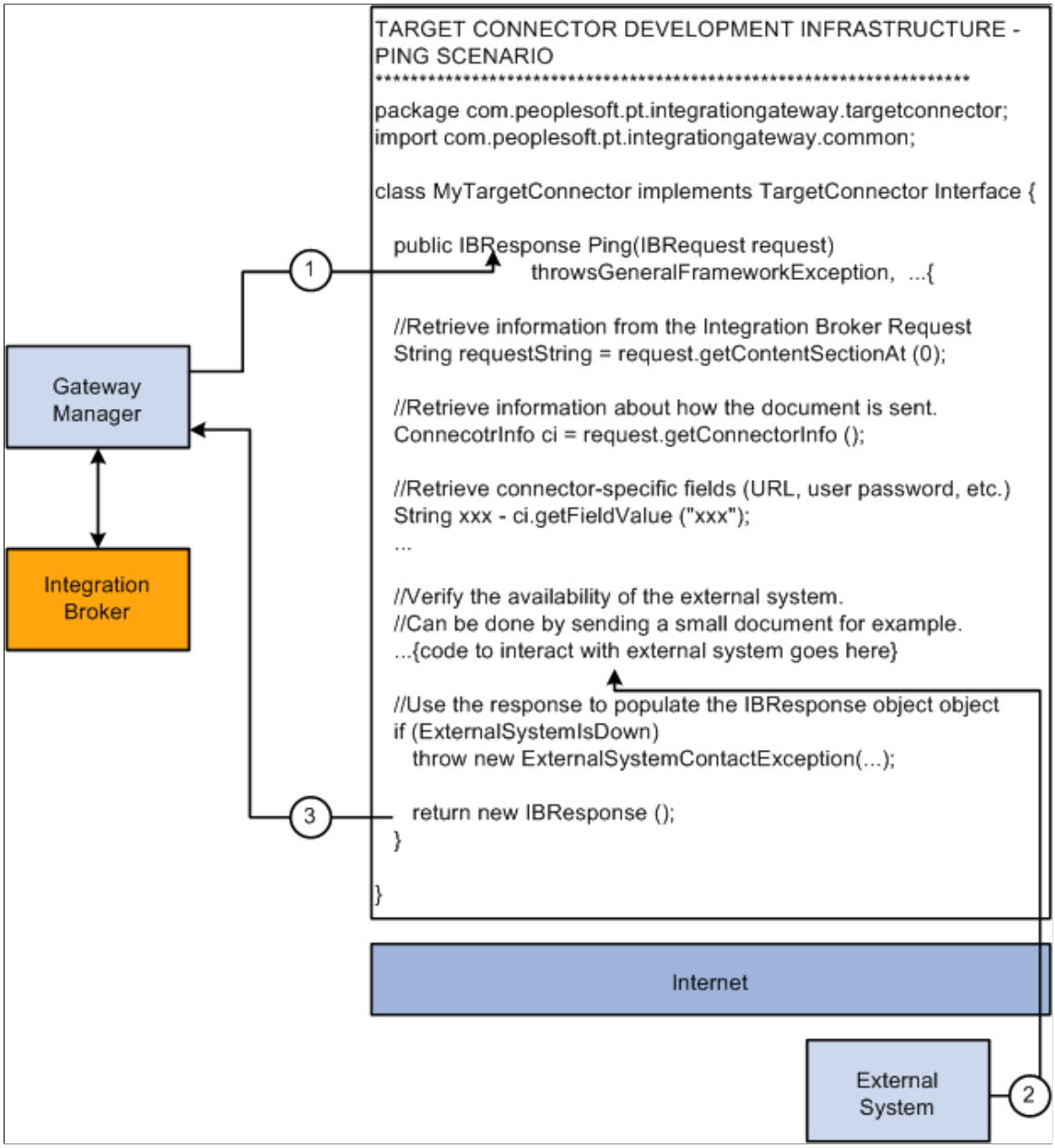
The following diagram shows how the Send method connector code generates and sends message requests to integration participants and returns responses:

This diagram shows how the Send method connector code generates and sends message requests to integration participants and returns responses.



The following diagram shows how the Ping method connector code pings external systems.

This diagram shows how the Ping method connector code pings external systems.



ConnectorDataCollection invokes introspection and the introspectConnector method is used by the application server to discover the connector properties that are used with the given target connector.

Building Introspection into Target Connectors

PeopleSoft Integration Broker can introspect (query) the capabilities of target connectors that are installed on a local or remote integration gateway by using introspection. Load all target connectors that are delivered with PeopleSoft Integration Broker by clicking the Load button on the Connectors page in the Gateways component.

You can build introspection into custom-built connectors. When you do so, you can load the connector and its properties with the click of a button.

For the introspection process to gather information about a custom target connector, you must implement the IntrospectConnector method.

The following example shows the connector properties that are available for use with the SMTP target connector:

```
public ConnectorDataCollection introspectConnector() {
   //Creates the ConnectorDataCollection that will be returned
   //by this method. This object will contain all the
   //necessary information about this Connector's properties.

      ConnectorDataCollection conCollection = new ConnectorDataCollection();

      //Create ConnectorData Object and stipulating the name of
      //the connector as seen from the Gateway Component.

      ConnectorData conData = new ConnectorData("SMTPTARGET");

      conData.addConnectorField("DestEmailAddress", true, "", "");
      conData.addConnectorField("SourceEmailAddress", true, "", "");
      conData.addConnectorField("CC", false, "", "");
      conData.addConnectorField("BCC", false, "", "");
      conData.addConnectorField("HEADER", Content-type”, false,
         "", "text/plain|text/html");
      conData.addConnectorField("HEADER","sendUncompressed",true,
         "Y","Y|N");

      //Add the ConnectorData to your ConnectorDataCollection
      //Object. Typically, you would only
      //add one ConnectorData into your ConnectorDataCollection.
      conCollection.addConnectorData(conData);

   return conCollection;
}
```

Use the addConnectorField method to add connector fields:

addConnectorField (*[PropertyID] PropertyName*, *Required*, *DefaultValue*, *PossibleValues*)

Use the following parameters for this method:

Parameter	***Description***
Property ID	Identifies different property types, such as HEADER for HTTP or SMTP. PeopleSoft software also uses the HEADER property ID to allow a message to be sent in either compressed or clear format through the sendUncompressed property. If this parameter is not supplied, the property ID is the connector name.
Property Name	Identifies the name of the connector property.
Required	Determines whether the information is required for the target connector to deliver its message. Values are: • *Y:* True • *N:*False

Parameter	*Description*
Default Value	Identifies the default value for the property. Typically, you set the Required parameter to True when you specify a default value so that this information carries to the node configuration in the integration engine.
Possible Values	Identifies a list of the possible values that the property can take, separated by the | character.

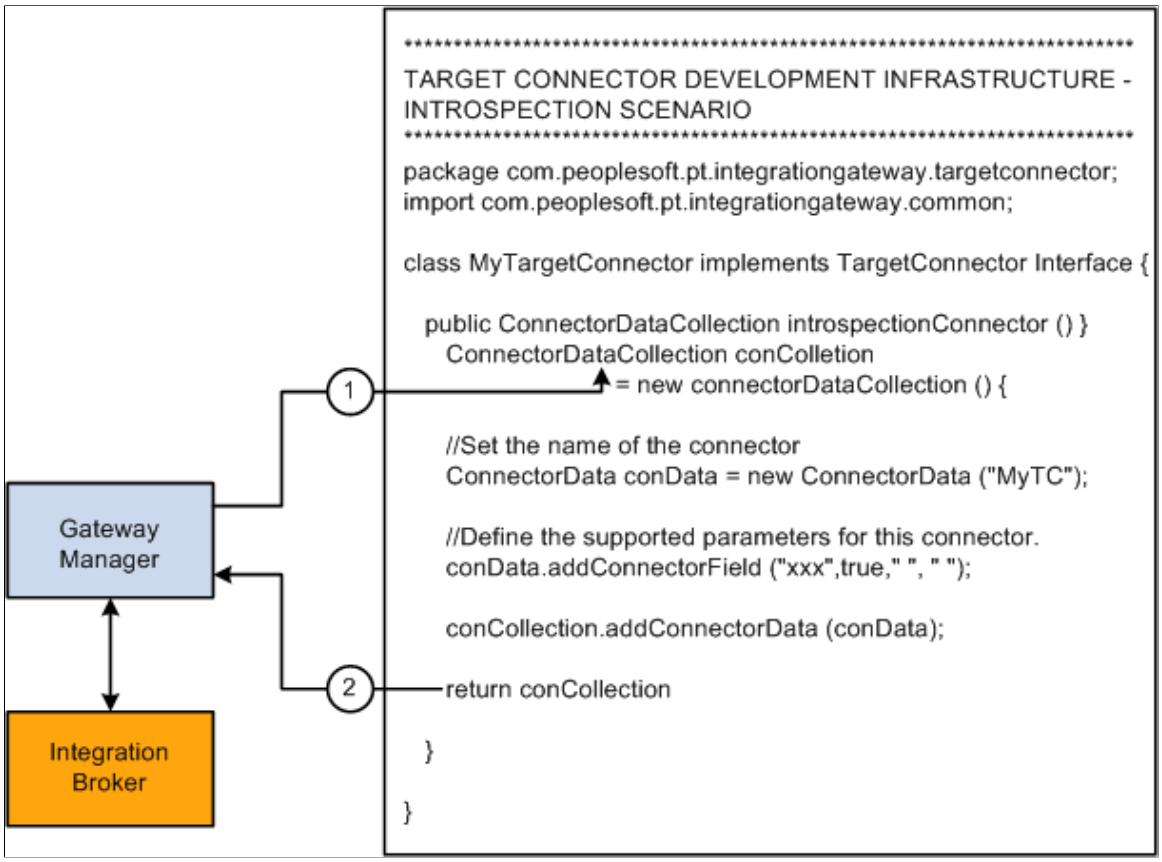
The following definition shows how these properties function:

```
conData.addConnectorField("HEADER","sendUncompressed",true,"Y, "Y|N");
```

In this case, the property name is sendUncompressed and its property ID is HEADER. The sendUncompressed property is required (the third parameter is set to true), so that whenever you create a node in the node definition component and specify SMTPTARGET as the target connector, this property appears on the page automatically. Further, because the default value is set to *Y*, this is the default value. Possible values have been identified as *Y* or *N*. If you click the list box (search box) for this property on the Connectors tab in the Node Definition component, you can select from those two values.

The following diagram shows how connector code accomplishes introspection.

This diagram shows how connector code accomplishes introspection.



Building Error Handling and Logging into Target Connectors

The following code example demonstrates how to build error handling and logging into target connectors:

```
package com.peoplesoft.pt.integrationgateway.targetconnector;

import ...

public class SampleTargetConnector implements TargetConnector {
         public IBResponse ping(IBRequest request)

         public IBResponse send(IBRequest request)throws
                     GeneralFrameworkException,
                     InvalidMessageException,
                     ExternalSystemContactException,
                     ExternalApplicationException,
                     MessageMarshallingException,
                     MessageUnmarshallingException,
                     DuplicateMessageException  {

      PSHttp httpObj = null;
      try {

      // Get handle on rootnode
      XmlNode root = dom.GetDocumentElement();

      // Cast the IBRequest back to an InternalIBRequest
      InternalIBRequest request = (InternalIBRequest)requestOrig;

      // Populate App Msg XML  Dom Object from IBRequest

      ...

      // Get the URL from either the IBRequest or from the
      //prop file (default)

      String URL = request.getConnectorInfo().getFieldValue("URL");

      // Log the request
      Logger.logMessage("SampleTargetConnector:
      Application Message Request", dom.GenerateXmlString(),
      Logger.STANDARD_INFORMATION);

      // Send the request DOM Document
      httpObj.setRequestProperty("content-type", "text/plain");
      httpObj.send(dom.GenerateXmlString().getBytes());

      // Get the response and convert to a String
      responseString = new String(httpObj.getContent());

      // Log the response
      Logger.logMessage("SampleTargetConnector:
      Application Message Response", responseString,
      Logger.STANDARD_INFORMATION);

      // Construct the IBResponse
      response = new IBResponse();

      ...

      // Return the successful IBResponse
      return response;

   } catch (XmlException xe) {
      httpObj.disconnect();
      throw new GeneralFrameworkException ("SampleTargetConnector:Failed
         while parsing XML");

   } catch (org.w3c.www.protocol.http.HttpException httpe) {
      throw new ("SampleTargetConnector:HTTP Protocol
```



```
            exception",httpe);

      } catch (java.io.IOException ioe) {
         throw new ExternalSystemContactException
            ("SampleTargetConnector:I/O Exception",ioe);

      } finally {
         httpObj.disconnect();
      }
      } // end send()
}
```

Developing Listening Connector Classes

This section discusses listening connector class development and discusses how to:

- Build servlet-based and nonservlet-based listening connectors.
- Invoke listening connectors.
- Control message routing.
- Build error handling and logging into listening connectors.

Building Servlet-Based and Nonservlet-Based Listening Connectors

If you require a listening connector that services HTTP requests, build a servlet-based listening connector. A servlet-based listening connector runs in the Servlet container on the web server.

See SDK Connector Examples.

This PeopleBook does not discuss how to install servlets on web servers.

See The servlet documentation for your web server.

Invoking Listening Connectors

Listening connectors must invoke PeopleSoft Integration Broker through the gateway manager Connect method.

```
IBResponse connect(IBRequest) throws
   GeneralFrameworkException
   DuplicateMessageException
   InvalidMessageException
   MessageMarshallingException
   MessageUnmarshallingException
   ExternalSystemContactException
   ExternalApplicationException
```

Controlling Message Routing

By accessing and modifying key information on the IBRequest, you can control the behavior of transactions as they flow through the integration gateway.

This section describes several dispatching features that you can use to control message routing by modifying the IBRequest from the listening connector, including routing messages to:

- Other (remote) integration gateways.
- Specific target connectors.
- Other PeopleSoft systems.

You can control the routing of a message to another integration gateway by specifying the uniform resource locator (URL) of the gateway in the IBRequest. You might need to forward messages to another gateway so that they can be processed by a remote PeopleSoft Integration Broker system. To do so, specify the URL of this integration gateway as follows:

```
. . .
IBRequest ibRequest = new IBRequest();
IbRequest.setOperationName("RemoteRoutingTest");
IbRequest.setRequestingNode("SourceSystem");
IbRequest.setPassword("myPassword");
. . .

//Specify the processing of the message to occur from
//anotherIntegration Gateway.
ibRequest.setRemoteFrameworkURL("https://hostName/PSIGW/
PeopleSoftListeningConnector");
```

You can also route a message to a specific target connector by modifying the request's ConnectorInfo object as follows:

```
. . .
IBRequest ibRequest = new IBRequest();
. . .

// Send a message through the HttpTargetConnector for example.
ConnectorInfo connectorInfo = ibRequest.getConnectorInfo();

connectorInfo.setConnectorClassName("HttpTargetConnector");
connectorInfo.setField("URL","http://www.externalsite.com");
connectorInfo.setField("Method","POST");
. . .
```

Building Error Handling and Logging into Listening Connectors

This is sample code for building error handling and logging into listening connectors:

```
package com.peoplesoft.pt.integrationgateway.listeningconnector;

import ...

public class HttpListeningConnector extends HttpServlet {
      public void doGet(HttpServletRequest req,
      HttpServletResponse resp) throws ServletException, IOException {
      }

      public void doPost(HttpServletRequest req,
      HttpServletResponse resp) throws ServletException, IOException {

         String actualResponse ="";
         IBRequest request = null;
         IBResponse response = null;

      try {

         String inputString = MiscUtil.readerToString(new
         InputStreamReader(req.getInputStream()));

         // Log the actual Input String
         Logger.logMessage("HttpListeningConnector: HTTP
```



```
            Request", inputString, Logger.STANDARD_INFORMATION);

            HttpListeningConnectorUtility util = new
            HttpListeningConnectorUtility();
            request = util.createIBRequest("XML", req, inputString);

            // Use the GatewayManager to invoke the Integration
            // Server and return its response.
            GatewayManager conMgr = new GatewayManager();
            response = conMgr.connect(request);

            // Need to get the actual response from the
            //IBResponse
            actualResponse = response.getContentSectionAt(0);

      } catch (InvalidMessageException ime) {
            ime.printStackTrace();
            actualResponse = getErrorXml(ime);
            Logger.logError("HTTPListeningConnector:
            InvalidMessageException", request, response,
            Logger.STANDARD_GATEWAY_EXCEPTION, ime);

      } catch (ExternalSystemContactException esce) {
            esce.printStackTrace();
            actualResponse = getErrorXml(esce);
            Logger.logError("HTTPListeningConnector:
            ExternalSystemContactException", request, response,
            Logger.STANDARD_GATEWAY_EXCEPTION, esce);

      } catch (ExternalApplicationException esee) {
         esee.printStackTrace();
         actualResponse = getErrorXml(esee);
         Logger.logError("HTTPListeningConnector:
         ExternalApplicationException", request, response,
         Logger.STANDARD_GATEWAY_EXCEPTION, esee);

      } catch (MessageMarshallingException mme) {
         mme.printStackTrace();
         actualResponse = getErrorXml(mme);
         Logger.logError("HTTPListeningConnector:
         MessageMarshallingException", request, response,
         Logger.STANDARD_GATEWAY_EXCEPTION, mme);

      } catch (MessageUnmarshallingException mue) {
         mue.printStackTrace();
         actualResponse = getErrorXml(mue);
         Logger.logError("HTTPListeningConnector:
         MessageUnmarshallingException", request, response,
         Logger.STANDARD_GATEWAY_EXCEPTION, mue);

      } catch (GeneralFrameworkException gfe)  {
         gfe.printStackTrace();
         actualResponse = getErrorXml(gfe);
         Logger.logError("HTTPListeningConnector:
         GeneralFrameworkException", request, response,
         Logger.STANDARD_GATEWAY_EXCEPTION, gfe);
      }

      // Return the message to the original requestor that
      //invoked the Servlet
      HttpListeningConnectorUtility.
            sendResponseBackToRequestor(actualResponse, resp);

      // Log the actual output String
      Logger.logMessage("HttpListeningConnector:
         HTTP Response", actualResponse, Logger.STANDARD_INFORMATION);

      }   // end doPost()
   }
```


Installing Connector Classes

Install connector classes on the local web server.

Installing Target Connector Classes

To install a target connector class, copy the class from the Java Classes directory to the following location on the local web server:

<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF\classes\com\peoplesoft\pt\integrationgateway\targetconnector

Installing Listening Connector Classes

To install a listening connector class, copy the class to the following location on the local web server:

<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF\classes\com\peoplesoft\pt\integrationgateway\listeningconnector

Registering Connectors

Before you can use a target connector, you must register it on the integration engine. To register a connector, load the connector information in the Gateways component by using the Load button. Loading the connector makes its capabilities known to PeopleSoft Integration Broker.

Then, assign the connector to the intended node on the Connector page in the Node Definition component. Enter the connector ID that corresponds to the new connector and edit the properties, as needed.

Servlet based listening connectors need to be registered with the web server. The mechanism for doing so is web server-specific. For Weblogic, this involves adding entries for the servlet and mapping to the web.xml file found in the following location:

```
<PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PSIGW.war\WEB-INF
```

Related Links

Loading Target Connectors