ORACLE

PEOPLESOFT

PeopleSoft 9.2: Integration Interfaces

July 2013

ORACLE

PeopleSoft 9.2: Integration Interfaces
CDSKU fscm92pbr1_r03_hosted-doc

Trademark Notice

License Restrictions Warranty/Consequential Damages Disclaimer

Warranty Disclaimer

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

Restricted Rights Notice

Hazardous Applications Notice

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

Third Party Content, Products, and Services Disclaimer

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

Alpha and Beta Draft Documentation Notice

If this document is in preproduction status:

This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

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.

PeopleSoft Hosted Documentation

You access the PeopleSoft Online Help on Oracle's PeopleSoft Hosted Documentation website, which enables you to access the full help website and context-sensitive help directly from an Oracle hosted server. The hosted documentation 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, because that documentation is now incorporated into the hosted website content. The Hosted Documentation website is available in English only.

Locally Installed Help

If your organization has firewall restrictions that prevent you from using the Hosted Documentation website, you can install the PeopleSoft Online Help locally. If you install the help locally, you have more control over which documents users can access and you can include links to your organization's custom documentation on help pages.

In addition, if you locally install the PeopleSoft Online Help, you can use any search engine for full-text searching. Your installation documentation includes instructions about how to set up Oracle Secure Enterprise Search for full-text searching.

See *PeopleTools 8.53 Installation* for your database platform, "Installing PeopleSoft Online Help." If you do not use Secure Enterprise Search, see the documentation for your chosen search engine.

Note: Before users can access the search engine on a locally installed help website, you must enable the Search portlet and link. Click the Help link on any page in the PeopleSoft Online Help for instructions.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format. 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 lines 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 line. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, 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: PeopleSoft 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***
Bold	Highlights PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Highlights field values, emphasis, and PeopleSoft or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. Italics also highlight references to words or letters, as in the following example: Enter the letter *O*.
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.
`Monospace font`	Highlights a PeopleCode program or other code example.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.

Typographical Convention	***Description***
{ } (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.
⇒	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)

Access to Oracle Support

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

Documentation Accessibility

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

Using and Managing the PeopleSoft Online Help

Click the Help link in the universal navigation header of any page in the PeopleSoft Online Help to see information on the following topics:

- What's new in the PeopleSoft Online Help.
- PeopleSoft Online Help acessibility.
- Accessing, navigating, and searching the PeopleSoft Online Help.
- Managing a locally installed PeopleSoft Online Help website.

PeopleSoft Enterprise Components Related Links

PeopleSoft Information Portal on Oracle.com

My Oracle Support

PeopleSoft Training from Oracle University

Contact Us

Send us your suggestions Please include release numbers for the PeopleTools and applications that you are using.

Follow Us

Get the latest PeopleSoft updates on Facebook.

Follow PeopleSoft on Twitter@PeopleSoft_Info.

Chapter 1

Setting Up the Credit Card Interface

Understanding XML-Based Integration Using Integration Broker

You can use PeopleSoft Integration Broker to process your PeopleSoft credit card transactions with the third-party credit card processing vendor of your choice. Oracle delivers messages and a synchronous service for you to use to set up a synchronous connection with your third-party vendor. Because each third-party vendor requires a different format for communication, PeopleSoft applications do not deliver transformations to your third-party vendor. This is because each third-party vendor has different requirements for inputs and outputs to their services. Therefore, part of setting up an integration involves writing your own transformation programs for the request and response message. The transformation programs are a mapping between the fields in the PeopleSoft system and the fields in your third-party vendor's service. You also need to set up a node in the PeopleSoft application to process the response from your third-party vendor.

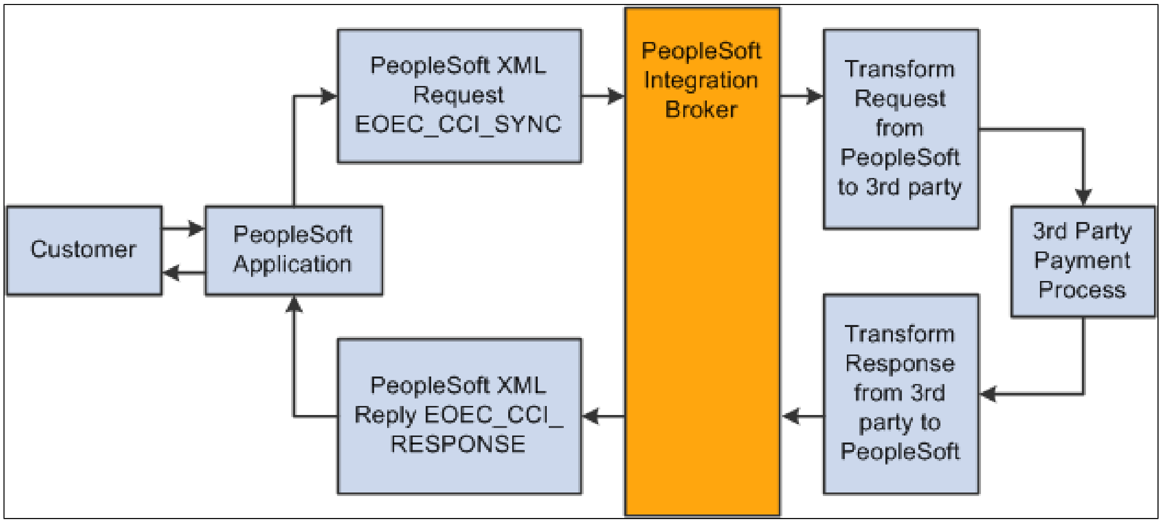
Oracle delivers XML messages for use with XML-based credit card processing vendors. You must build your own XML message transformation into the format that the vendor is expecting.

You can use PeopleSoft Application Engine to perform your transformations. You use the delivered EOEC_CCI_SYNC message for the request transaction and the delivered EOEC_CCI_RESPONSE message for the response transaction. The messages are detailed later in this section.

Note: If you have upgraded from a PeopleTools 8.47 or earlier release, the upgrade program creates service operations for these messages. The service operation names and message names are the same.

Image: EOEC_CCI_SYNC and EOEC_CCI_RESPONSE is used to integrate with a third-party credit processing vendor

The diagram shows the process flow for integration with a third-party credit processing vendor.


Customer
PeopleSoft Application
PeopleSoft XML Request EOEC_CCI_SYNC
PeopleSoft Integration Broker
Transform Request from PeopleSoft to 3rd party
3rd Party Payment Process
Transform Response from 3rd party to PeopleSoft
PeopleSoft XML Reply EOEC_CCI_ RESPONSE

EOEC_CCI_Sync Message

The EOEC_CCI_SYNC message is a synchronous request that the credit card interface sends to the third-party vendor. The request can be for an authorize, bill, authorize and bill, credit transaction, or authorization reversal. The PeopleCode that supports this message is located in App Package EOEC_CCI. The following tables describe the request fields and how they are populated by the PeopleSoft system.

Level 0 Record: EOEC_CCI_XMLPAY:

Message field	***Alias***	***Populated with***
EOEC_CCI_UNIQUE_ID	UNIQUEID	Unique ID generated for each transaction.

Level 1 Record: EOEC_CCI_RQST:

Message field	***Alias***	***Populated with***
EOEC_CCI_MERCHANT	VENDOR	Populated from the merchant ID set up on the Credit Card Interface Installation page.
EOEC_CCI_PARTNER	PARTNER	Hardcoded to “PeopleSoft”.

Level 2 Record: EOEC_CCI_TRANS:

Message field	***Alias***	***Populated with***
EOEC_CCI_TRANSID	TRANSACTION_ID	Either blank or contains the request ID of a previous transaction (such as a prior authorization transaction).
EOEC_CCI_MERCH_REF	TRANSACTION_CUSTREF	Contains a reference to the transaction such as an order or invoice number.
EOEC_CCI_RQSTTOKEN	N/A	Either blank or contains the request token of a previous transaction (such as a prior authorization transaction).

Level 3 Record: EOEC_CCI_TRNTYP:

Message field	***Alias***	***Populated with***
EOEC_CCI_TRANSACT	TRANSTYPE	The service to be performed: • 1=Authorize Only • 2=Authorize & Bill • 3=Bill Only • 4=Credit • 5=Authorize Reversal

Level 4 Record: EOEC_CCI_PAYDAT:

Message field	***Alias***	***Populated with***
EOEC_CCI_TRANSACT	N/A	N/A

Level 5 Record: EOEC_CCI_INV:

Message field	***Alias***	***Populated with***
EOEC_CCI_INV_NUM	INVNUM	Either blank or contains the request ID of a pervious transaction (such as a prior authorization transaction). Oracle recommends that you use TRANSACTION_ID instead of INVNUM in transformation programs because the field is only 20.
EOEC_DATE		N/A
DESCR254	DESCRIPTION	Hardcoded to “Description”.
EOEC_CCI_DISC_AMT	DISCOUNTAMT	N/A
EOEC_CCI_SHIP_AMT	SHIPAMT	N/A
EOEC_CCI_DUTY_AMT	DUTYAMT	N/A
EOEC_CCI_TAX_AMT	TAXAMT	N/A
EOEC_CCI_TAX_INCL	NATIONALTAXINCL	N/A
EOEC_CCI_TOTAL_AMT	TOTALAMT	The total amount of the transaction.
EOEC_CCI_COMMENT	COMMENT	N/A
CURRENCY_CD	N/A	N/A

Level 6 Record: EOEC_CCI_BILLFM:

Message field	***Alias***	***Populated with***
EOEC_CCI_FULLNAME	NAME	N/A
EOEC_EMAIL_ADDR	EMAIL	N/A
PHONE	N/A	N/A
FAX	N/A	N/A
URL	N/A	N/A

Level 7 Record: EOEC_CCI_ADDR1:

Message field	***Alias***	***Populated with***
ADDRESS1	STREET	N/A
ADDRESS2	N/A	N/A
ADDRESS3	N/A	N/A
ADDRESS4	N/A	N/A
CITY	N/A	N/A
STATE	N/A	N/A
POSTAL	ZIP	N/A
COUNTRY	N/A	N/A

Level 6 Record: EOEC_CCI_BILLTO:

Message field	***Alias***	***Populated with***
EOEC_CCI_CUSTID	CUSTOMERID	N/A
EOEC_CCI_FULLNAME	NAME	N/A
EOEC_EMAIL_ADDR	EMAIL	Email address of the consumer.
PHONE	N/A	Telephone number of the consumer.
FAX	N/A	N/A
EOEC_CCI_CUSTCODE	CUSTCODE	N/A

Message field	***Alias***	***Populated with***
EOEC_CCI_PO_NUM	PONUM	N/A
EOEC_CCI_TAXEXEMPT	TAXEXEMPT	N/A

Level 7 Record: EOEC_CCI_ADDR2:

Message field	***Alias***	***Populated with***
ADDRESS1	STREET	Street address of the consumer.
ADDRESS2	N/A	N/A
ADDRESS3	N/A	N/A
ADDRESS4	N/A	N/A
CITY	N/A	City address of the consumer.
STATE	N/A	State address of the consumer.
POSTAL	ZIP	Postal address of the consumer.
COUNTRY	N/A	Country address of the consumer.

Level 6 Record: EOEC_CCI_SHIPFM:

Message field	***Alias***	***Populated with***
EOEC_CCI_FULLNAME	NAME	N/A
EOEC_EMAIL_ADDR	EMAIL	N/A
PHONE	N/A	N/A
FAX	N/A	N/A

Level 7 Record: EOEC_CCI_ADDR3:

Message field	***Alias***	***Populated with***
ADDRESS1	STREET	N/A
ADDRESS2	N/A	N/A
ADDRESS3	N/A	N/A
ADDRESS4	N/A	N/A

Message field	***Alias***	***Populated with***
CITY	N/A	N/A
STATE	N/A	N/A
POSTAL	ZIP	N/A
COUNTRY	N/A	N/A

Level 6 Record: EOEC_CCI_SHIPTO:

Message field	***Alias***	***Populated with***
EOEC_CCI_FULLNAME	NAME	N/A
EOEC_EMAIL_ADDR	EMAIL	N/A
PHONE	N/A	N/A
FAX	N/A	N/A

Level 7 Record: EOEC_CCI_ADDR4:

Message field	***Alias***	***Populated with***
ADDRESS1	STREET	N/A
ADDRESS2	N/A	N/A
ADDRESS3	N/A	N/A
ADDRESS4	N/A	N/A
CITY	N/A	N/A
STATE	N/A	N/A
POSTAL	ZIP	N/A
COUNTRY	N/A	N/A

Level 6 Record: EOEC_CCI_ITEM:

Message field	***Alias***	***Populated with***
EOEC_CCI_ITEM NUM	ITEM_NUMBER	N/A
EOEC_CCI_SKU	SKU	N/A

Message field	***Alias***	***Populated with***
EOEC_CCI_UPC	UPC	N/A
DESCR254	DESCRIPTION	N/A
EOEC_CCI_QTY AMT TOTALAMT	QUANTITY	N/A
EOEC_CCI_UOM	UNITOFMEASURE	N/A
EOEC_CCI_UNITPRICE	UNITPRICE	N/A
EOEC__CCI_EXTAMT	EXTAMT	N/A
EOEC_CCI_DISC_AMT	DISCOUNTAMT	N/A
EOEC_CCI_TAX_AMT	TAXAMT _	N/A
EOEC_CCI_TOTAL	TOTALAMT	N/A

Level 4 Record: EOEC_CCI_TENDER:

Message field	***Alias***	***Populated with***
EOEC_CCI_TRANSACT	N/A	N/A

Level 5 Record: EOEC_CCI_CARD:

Message field	***Alias***	***Populated with***
EOEC_CCI_TYPE	CARDTYPE	Two-character code for the type of card used in the transaction. • 01=Visa • 02=MasterCard • 03=Diners Club • 04=American Express • 05=Discover
EOEC_CCI_NUMBER	CARDNUM	Credit card number used in the transaction.
EOEC_CCI_EXPYR	EXPYR	Expiration year of the card.
EOEC_CCI_EXPMO	EXPMO	Expiration month of the card.
EOEC_CCI_CVNUM	CVNUM	Card verification number.

Message field	***Alias***	***Populated with***
EOEC_CCI_MAGDATA	MAGDATA	N/A
EOEC_CCI_FULLNAME	NAMEONCARD	First and last name of the consumer.
EOEC_CCI_FNAME	FIRSTNAME	First name of the consumer.
EOEC_CCI_LNAME	LASTNAME	Last name of the consumer.

Level 5 Record: EOEC_CCI_STATUS:

Message field	***Alias***	***Populated with***
EOEC_CCI_TRANS_REF	PNREF	N/A

Note: When writing your transformation program, use the alias name to reference the fields. When you view the “Request — Original” text of the message, the alias name is displayed.

EOEC_CCI_RESPONSE Message

The EOEC_CCI_RESPONSE message is a response to the request that the credit card interface receives from the third-party vendor. Your transformation should populate the response message fields as shown in the tables.

Level 0 Record: EOEC_CCI_XMLRSP:

Message field	***Alias***	***Populate with***
EOEC_CCI_UNIQUE_ID	UNIQUEID	N/A

Level 1 Record: EOEC_CCI_RSPNS:

Message field	***Alias***	***Populate with***
EOEC_CCI_MERCHANT	VENDOR	N/A
EOEC_CCI_PARTNER	PARTNER	N/A

Level 2 Record: EOEC_CCI_TRRSLT:

Message field	***Alias***	***Populate with***
EOEC_CCI_TRANSID	TRANSACTIONID	The request ID or identifier returned from the third-party vendor.

Message field	***Alias***	***Populate with***
EOEC_CCI_RESULT	RESULT	Map the appropriate return code for the transmission result in the following way: • 0=Approve (no error) • 100-199=Decline For self-service application, you can display the vendor return message to self-service users. • 200-299=Decline For self-service application, do not display the vendor return message to self-service users. • 300-399=Connection Error
EOEC_CCI_AVS	AVS	N/A
EOEC_CCI_CVRESULT	CVRESULT	N/A
EOEC_CCI_RET_MSG	MESSAGE	Populate this with a text response from the vendor or this can be populated with text that explains to the user why a transaction was not successful.
EOEC_CCI_TRANS_REF	PNREF	N/A
EOEC_RET_AUTHCD	AUTHCODE	This should be populated with the authorization code returned from the vendor.
EOEC_CCI_HOSTCODE	HOSTCODE	N/A
EOEC_CCI_HOST_URL	HOSTURL	N/A
EOEC_CCI_ORIGRSLT	ORIGRESULT	N/A
EOEC_RET_STATUS	TRSTATUS	N/A
EOEC_RET_STATUSMSG	N/A	N/A
EOEC_RET_AUTHDTTM	N/A	N/A
EOEC_CCI_RQSTTOKEN	N/A	N/A

Level 3 Record: EOEC_CCI_AVRSLT

Message field	***Alias***	***Populate with***
EOEC_MATCH_STREET	STREETMATCH	N/A
EOEC_MATCH_ZIP	ZIPMATCH	N/A

When writing the transformation program, use the alias name to reference the fields.

Note: The alias name is shown in the “Response — Original” message from within the Service Operations Monitor.

Agents can then process credit cards using their application-specific credit card transaction page to submit the transaction to the vendor for authorization, billing, authorization and billing, or credit. You can choose which types of transactions to permit.

For more information, refer the product documentations for *PeopleTools: PeopleSoft Integration Broker*

PeopleTools: Security Administration

PeopleTools: Portal Technology.

Setting Up Credit Card Integration for Integration Broker

This section discusses how to configure integration for Integration Broker.

Configuring Integration for Integration Broker

To set up Integration Broker for credit card processing:

1. Define the Integration Broker Gateway if it's not already done.
2. Activate the delivered service (EOEC_CCI_SYNC) that is used for the credit card integration.
3. Set up an external node to use when using the XML-based interface to which the XML messages should be sent and to indicate where the processor is located.

   Also specify the authentication option that you arranged with the credit card processor.
4. Set up routings.

   Use the Routings tab in the Node component to add the EOEC_CCI_SYNC service to the node. To see the XML data before and after transformations, set the log detail to Header & Detail. This is helpful when troubleshooting your new integration.
5. Test the integration.

   You can use the test component described below.

If you receive error messages when using Integration Broker, see the Troubleshooting the Integration Broker Setup section below:

Troubleshooting the Integration Broker Setup

Several sources of information are available when the setup is not successful. These include but are not limited to:

- Error messages stored on the message instance (view with the message monitor).
- IB gateway error log (http://<GatewayMachine>:<Port>/PSIGW/errorLog.html).
- IB gateway message log (http://<GatewayMachine>:<Port>/PSIGW/msgLog.html).
- Application server log for the active IB domain server on the database.
- Web server logs.

You can also do the following:

- Increase the log fence on the gateway properties file. This file is located in the following directory: PS_CFG_HOME\webserv\<web-server>\applications\peoplesfoft\PSIGW

  If you set the log fence to five in the integration Gateway properties, you will receive more details in the error and message logs.

- Check the Service Operations Monitor to view the XML messages before and after the transformations (PeopleTools, Integration Broker, Monitor, Service Operations, Synchronous Services).
- Check the Header and Detail logs on the Synchronous Detail page (PeopleTools, Integration Broker, Service Operations Monitor, Monitor, Synchronous Details).

  Problems are usually due to incorrect transformations between the two systems. Use these logs to ensure that your transformations are correct.

  Your Response-Transformed message structure should look similar to this:

```
<?xml version="1.0"?>
<EOEC_CCI_RESPONSE xmlns:c="yourcompany.com">
  <FieldTypes>
    <EOEC_CCI_XMLRSP class="R"/>
    <EOEC_CCI_TRRSLT class="R">
      <RESULT type="NUMBER"/>
    </EOEC_CCI_TRRSLT>
    <EOEC_CCI_AVRSLT class="R"/>
    <EOEC_CCI_RSPNS class="R"/>
    <PSCAMA class="R"/>
  </FieldTypes>
  <MsgData>
    <Transaction>
      <EOEC_CCI_XMLRSP class="R">
        <UNIQUEID/>
        <EOEC_CCI_RSPNS class="R">
          <VENDOR/>
          <PARTNER/>
          <EOEC_CCI_TRRSLT class="R">
            <TRANSACTIONID>177</TRANSACTIONID>
            <RESULT>0</RESULT>
            <AVS/>
            <CVRESULT/>
            <MESSAGE/>
            <AUTHCODE>123456</AUTHCODE>
            <HOSTCODE/>
            <HOSTURL/>
            <ORIGRESULT/>
```

```
                <TRSTATUS/>
                <EOEC_RET_STATUSMSG/>
                <EOEC_RET_AUTHDTTM/>
                <EOEC_CCI_AVRSLT class="R">
                  <STREETMATCH/>
                </EOEC_CCI_AVRSLT>
              </EOEC_CCI_TRRSLT>
            </EOEC_CCI_RSPNS>
          </EOEC_CCI_XMLRSP>
        </Transaction>
      </MsgData>
    </EOEC_CCI_RESPONSE>
```

Setting Up Credit Card Interface Elements

This section discusses how to:

- Define connection parameters.
- Define accepted credit card types.
- Test the credit card interface.
- Test credit card transactions.

Note: The information in this section is used for credit card integration using Integration Broker, or any manual processing that you have set up for credit card processing vendors.

Pages Used to Set Up Credit Card Interface Elements

Page Name	***Definition Name***	***Navigation***	***Usage***
Credit Card Interface Installation	EOEC_CCI_INSTAL	Enterprise Components, Component Configurations, Credit Card Interface, Setup	Define connection parameters for credit card processing calls to a third-party vendor. Before you set up credit card processing options, establish your merchant account with a third-party vendor.
Card Type	EOEC_CCI_CARDTYPE	Enterprise Components, Component Configurations, Credit Card Interface, Credit Card Types	Define the types of credit cards you accept for credit card processing.
Test Credit Card Interface - Card Entry/Display	EOEC_CCI_TEST	Enterprise Components, Component Configurations, Credit Card Interface, Test Credit Card Interface, Card Entry/Display	Enter test credit card information that you can submit to verify that your credit card processing is functioning properly.

Page Name	***Definition Name***	***Navigation***	***Usage***
Test Credit Card Interface - Transaction	EOEC_CCI_TRANSACT	Enterprise Components, Component Configurations, Credit Card Interface, Test Credit Card Interface, Transaction	Enter test credit card transaction information that you can submit to verify that your credit card processing is functioning properly.

Credit Card Interface Installation Page

Use the Credit Card Interface Installation page (EOEC_CCI_INSTAL) to define connection parameters for credit card processing calls to a third-party vendor.

Before you set up credit card processing options, establish your merchant account with a third-party vendor.

Navigation

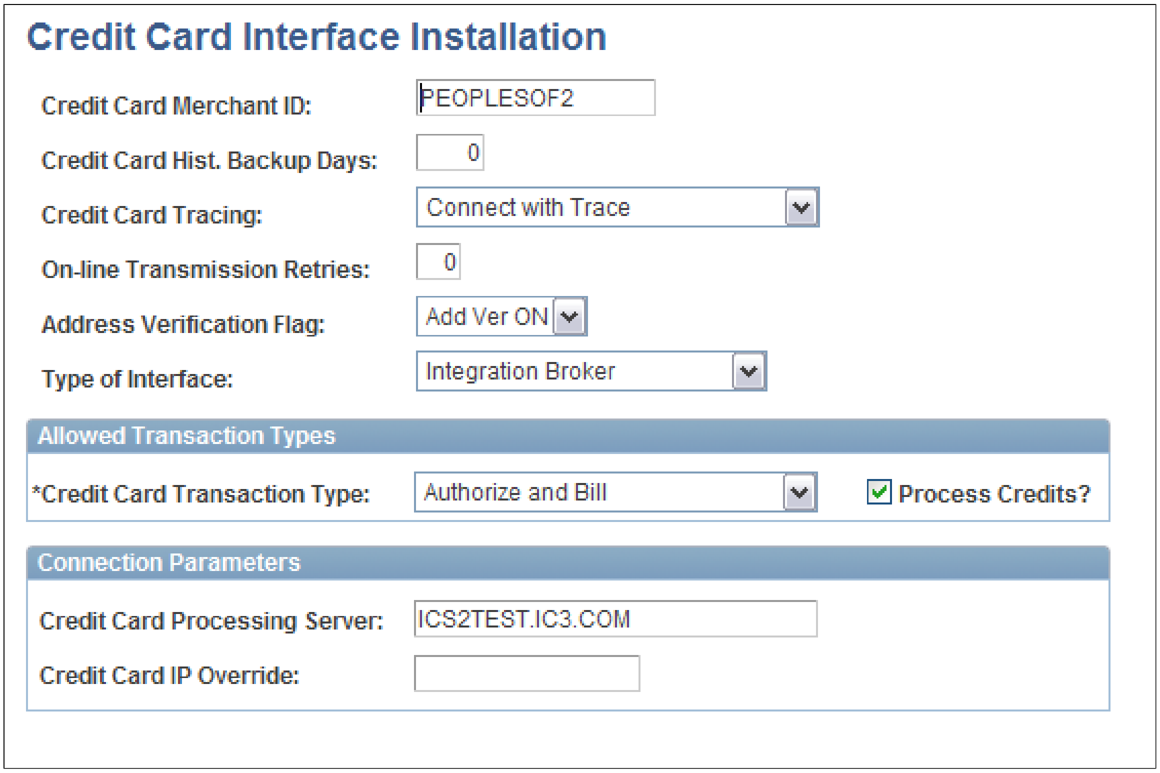
Enterprise Components, Component Configurations, Credit Card Interface, Setup

Note: Before you set up credit card processing options, establish your merchant account with a third-party vendor.

Note: Check the installation documentation for the product you are installing for specific details on setting up credit card interfaces.

Image: Credit Card Interface Installation page

This example illustrates the fields and controls on the Credit Card Interface Installation page. You can find definitions for the fields and controls later on this page.



Credit Card Interface Installation

Credit Card Merchant ID: PEOPLESOF2

Credit Card Hist. Backup Days: 0

Credit Card Tracing: Connect with Trace

On-line Transmission Retries: 0

Address Verification Flag: Add Ver ON

Type of Interface: Integration Broker

Allowed Transaction Types

*Credit Card Transaction Type: Authorize and Bill ☑ Process Credits?

Connection Parameters

Credit Card Processing Server: ICS2TEST.IC3.COM

Credit Card IP Override:

Verify connection requirements with your vendor.

Credit Card Merchant ID

Enter the merchant ID supplied by your vendor.

Credit Card Hist. Backup Days (credit card history backup days)

If you create a process that archives history records, specify the number of days that you retain credit card authorization history records.

Credit Card Tracing

This field is currently not used.

On-line Transmission Retries

Enter a value from *0* through *9* to specify how many times the system should attempt to retransmit transactions in the event of transmission failure.

Address Verification Flag

Credit card transmissions can fail authorization if the address that you send doesn't exactly match the billing address for the credit card. Select from:

Add Vf ON (address verification on): Transactions fail when the address that you send does not match the credit card billing address. This is the default value.

Add Vf OFF (address verification off): Transactions do not fail when the address that you send does not match the billing address on the credit card.

Type of Interface

Select to use *Integration Broker* as the interface type.

Allowed Transaction Types

Credit Card Transaction Type

Select the types of transactions that your agents are allowed to submit. Disallowed transaction types are not available on the application-specific credit card transaction page. Select from:

Authorization Reversal: Your vendor can cancel the transaction after authorization and before payment is received. This makes the funds available if the transaction is cancelled.

Authorize Only: Your vendor verifies that the card is valid for the charge. For example, the customer has enough credit to pay for the order, the card is not stolen, and so forth. The vendor does not bill the credit card.

Bill Only: Your vendor bills the card without first verifying that the card is valid for the charge. Select this option if you have preauthorized the transaction and you want to submit the transaction for billing only.

Authorize and Bill: Your vendor performs both authorization and billing during the same transaction. The vendor charges the customer's credit card on receiving authorization.

Credit Only: Your vendor credits the customer's credit card.

Process Credits

Select to permit agents to submit credit transactions as well as billing transactions. This option is available only if you selected either *Authorize and Bill* or *Bill Only* in the Credit Card Transaction Types field.

Connection Parameters

The third-party vendor that you integrate with will provide you with information to connect with their systems. Enter that information to enable your PeopleSoft system to make the connection when you submit a transaction for processing.

Card Type Page

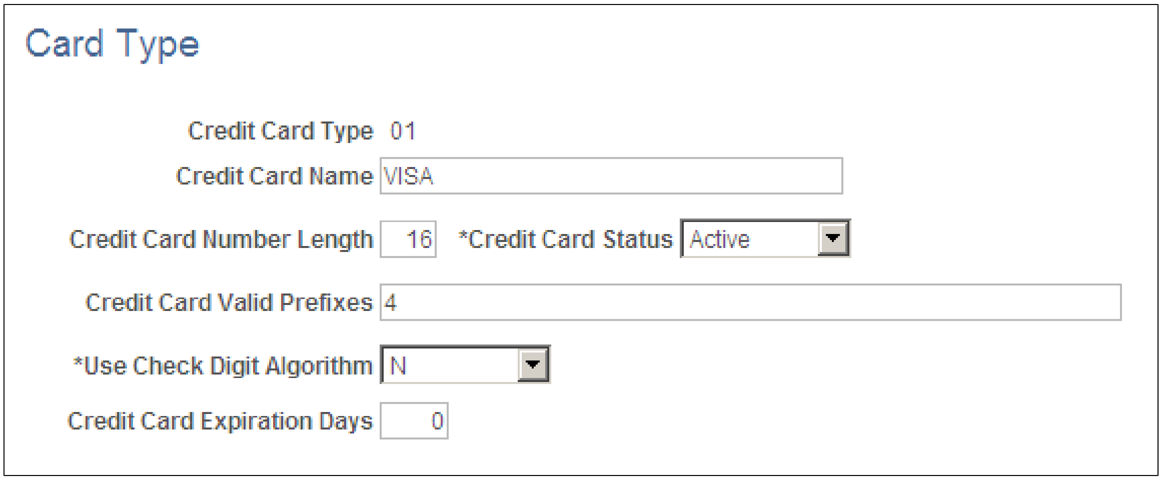
Use the Card Type page (EOEC_CCI_CARDTYPE) to define the types of credit cards you accept for credit card processing.

Navigation

Enterprise Components, Component Configurations, Credit Card Interface, Credit Card Types

Image: Card Type page

This example illustrates the fields and controls on the Card Type page. You can find definitions for the fields and controls later on this page.



Card Type

Credit Card Type 01

Credit Card Name VISA

Credit Card Number Length 16

*Credit Card Status Active

Credit Card Valid Prefixes 4

*Use Check Digit Algorithm N

Credit Card Expiration Days 0

Use this page to define the types of credit cards that you accept for credit card processing.

Oracle delivers data for most popular credit card types. You can modify existing definitions and add new ones.

Credit Card Type

Enter a value for the credit card.

Credit Card Name

Enter a credit card name such as Visa or MasterCard. The name should match the credit card type so that you can identify the card without memorizing the credit card type codes.

Credit Card Number Length

Enter the card's standard credit card number length. Before transmitting a request to your vendor, the system validates the length of the credit card number against this number.

Credit Card Status

Select *Active* if you accept this type of credit card. Select *Inactive* if you don’t accept this type of credit card. Inactive credit card types do not appear on the application-specific credit card transaction page or in the Test Credit Card Interface component.

The default value for this field is *Inactive.*

Credit Card Valid Prefixes

Enter all valid prefixes for this type of credit card. Enter multiple prefixes in comma-separated format with no spaces in between. The system removes any characters other than numbers and commas when you move to another field.

Before transmitting a request to your vendor, the system validates that the credit card number starts with a valid prefix.

Use Check Digit Algorithm

Select *Y* (yes) to use the modulus (MOD) 10 check digit algorithm to validate credit card numbers before transmitting requests to your vendor. The MOD 10 check digit algorithm verifies whether card numbers you enter into the system are legitimate.

The default value for this field is *N.*

Credit Card Expiration Days

Enter a four digit numerical expiration value.

Test Credit Card Interface - Card Entry/Display Page

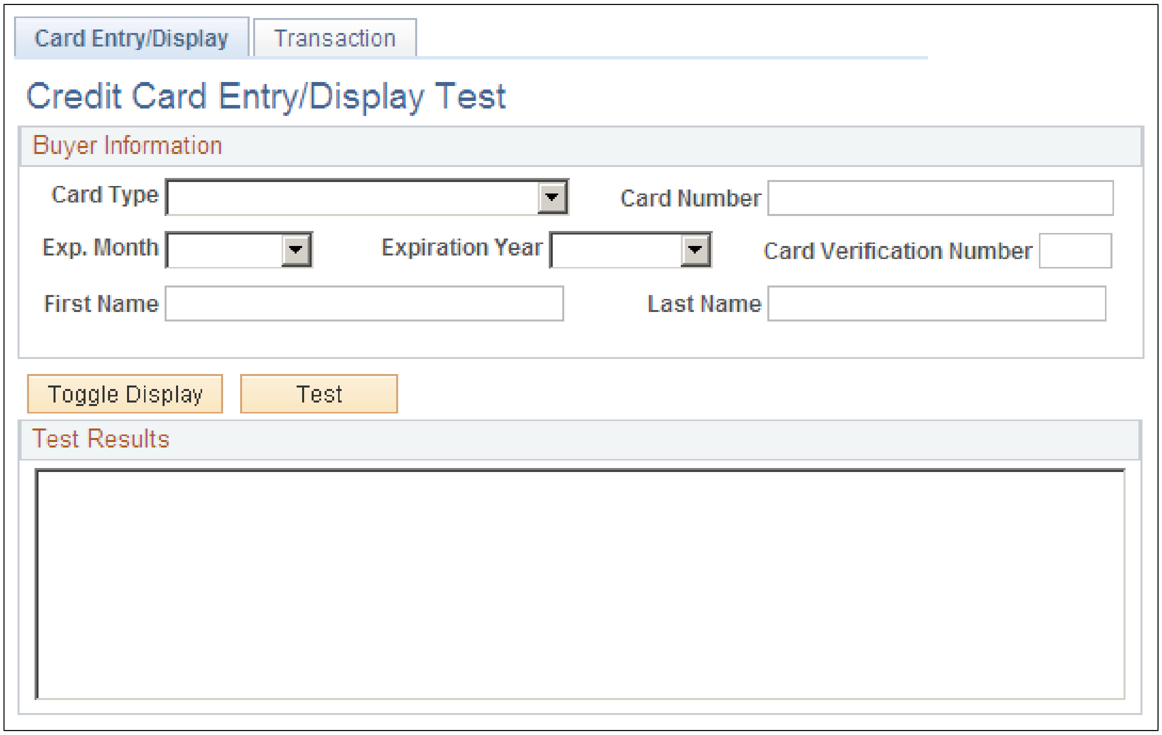
Use the Test Credit Card Interface - Card Entry/Display page (EOEC_CCI_TEST) to enter test credit card information that you can submit to verify that your credit card processing is functioning properly.

Navigation

Enterprise Components, Component Configurations, Credit Card Interface, Test Credit Card Interface, Card Entry/Display

Image: Test Credit Card Interface - Card Entry/Display page

This example illustrates the fields and controls on the Test Credit Card Interface - Card Entry/Display page. You can find definitions for the fields and controls later on this page.



Use this page to enter test credit card information that you can submit to verify that your credit card processing is functioning properly.

The test that you can run on this page:

- Verifies that the card number you enter meets the requirements defined in the Credit Card Valid Prefixes field for the associated card type on the Card Type page.
- If you have set the Use Check Digit Algorithm field value to *Y*, verifies that the card number is valid based on the MOD 10 check digit algorithm.
- Verifies that you have entered values in the Exp. Month, Expiration Year, First Name and Last Name fields on this page.

You can use the following credit card sample data in your test transactions:

Credit Card Type	***Credit Card Number***
American Express	378282246310005
Diners Club/Carte Blanche	38000000000006
Discover	6011111111111117
MasterCard	5555555555554444
Visa	4111111111111111

Card Type

Select a card type to test. Available values are defined on the Credit Card Types page.

Credit Card Number

Enter the credit card number to test.

Exp. Month (expiration month), Expiration Year, Card Verification Number, First Name and Last Name

Enter card information to test.

Card verification number is optional in running credit card tests.

Toggle Display

Click to switch between display-only and editable modes.

Test

Click to begin the test.

Test Results

The results of the test appear in this text box. If the card number is valid, the message VALID CARD NUMBER appears. If the card number is not valid, an explanatory message appears; the card number is incorrect or the card is expired, for example.

Credit Card Transaction Test Page

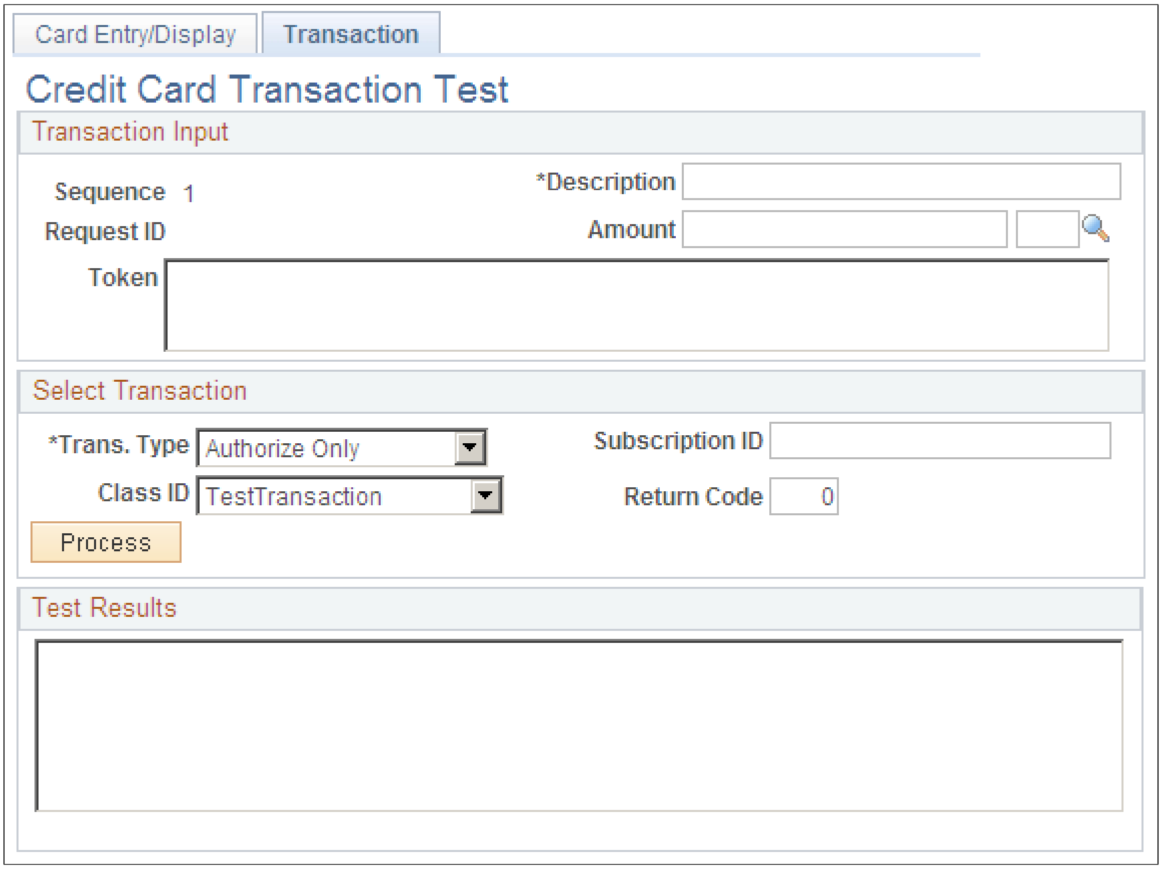
Use the Test Credit Card Interface - Transaction page (EOEC_CCI_TRANSACT) to enter test credit card transaction information that you can submit to verify that your credit card processing is functioning properly.

Navigation

Enterprise Components, Component Configurations, Credit Card Interface, Test Credit Card Interface, Transaction

Image: Test Credit Card Interface - Transaction page

This example illustrates the fields and controls on the Test Credit Card Interface - Transaction page. You can find definitions for the fields and controls later on this page.



Sequence 1

Request ID

*Description

Amount

Token

*Trans. Type: Authorize Only

Class ID: TestTransaction

Subscription ID

Return Code: 0

Use this page to enter test credit card transaction information that you can submit to verify that your credit card processing is functioning properly.

The test that you can run on this page verifies that your environment is set up correctly to process online credit card transactions using XML Compliant integration.

Note: This does not test the environment that is set up for batch credit card transactions.

Sequence and Request ID — Display a combination of numbers that distinguishes the transaction from other transactions. The combination of numbers are similar to a run control or job number.

Amount — Enter a transaction amount and click the Look up button to select a transaction currency.

Token

The token is a response value that is returned from the Credit Card processing service. It is another validation that can be referenced in regards to a specific transaction.

Trans. Type (transaction type)

Choose a transaction type to test:

Auth/Bill (Authorization/Bill): Perform both authorization and billing during the same transaction.

AuthRev (Authorization Reversal): Cancel the transaction after authorization and before payment is received.

Authorize: (default value) Verify that the card is valid for the charge.

Bill: Bill the card without first verifying that the card is valid for the charge.

Credit: Credit the customer’s credit card.

Class ID

Select *TestTransaction* (the default) or one of the following interface types that has been specified on the Credit Card Interface Installation page.

ProcessBrokerTransaction: Select if you want to test your XML-based interface.

InterlinkTransaction: Select if you want to test your Business Interlink interface.

Process

Click to process the test transaction.

Return Code

Enter a return code to test whether proper error messages and results are returned. Available codes and their descriptions are discussed in the Return Codes section subsequently.

Test Results

The results of the transaction test appear in this text box. Test result interpretations are discussed in the following sections.

Return Codes

You can enter any of the following return codes and click Process to view the corresponding description and error message in the Test Results area. These return codes and their corresponding error messages can appear in multiple areas. For example, when you are using the Test Credit Card Interface component, they appear on this test page. In an application, they appear as appropriate for that application's method of interacting with the credit card interface.

Return Code	***Description***
-3	Error Opening Trace File
-4	Vendor Error − ICS_INIT failed

Return Code	***Description***
-5	Unsupported Service
-6	Credit card number is invalid
-7	Phone number is too long
-8	State field length is invalid
-9	Zip Code field is too long
-10	Amount must be greater than zero
-11	Vendor Error − ICS_SEND failed
-12	Decryption Failed
-15	Request ID is required
-16	Currency is required
-17	Phone is required
-18	Email ID is required
-19	Zip Code is required
-20	City is required
-21	Country code is required
-23	Address 1 is required
-99	Trace Run Only

Chapter 2

Using PeopleSoft Directory Interface

Understanding PeopleSoft Directory Interface

PeopleSoft Directory Interface uses Lightweight Directory Access Protocol (LDAP) directory services to authenticate users of PeopleSoft applications.

PeopleSoft Directory Interface provides additional mappings and integration points, such as messages, that enable PeopleSoft data and LDAP data to stay synchronized. Most directory data, such as user ID, name, and email address, is also maintained in your PeopleSoft database. When you use PeopleSoft Directory Interface, you make selected PeopleSoft data available to the directory, and you maintain the data in the PeopleSoft database.

When information changes in the PeopleSoft database, PeopleSoft Directory Interface captures that updated information and automatically updates the equivalent information in the directory server, or it writes the updates to a file for you to apply at another time.

Understanding Data Mapping

PeopleSoft information is stored in tables according to a relational model. The information in your LDAP directory is stored in trees according to a hierarchical model. You use PeopleSoft Directory Interface to map selected PeopleSoft data to corresponding data in the directory service. When PeopleSoft Directory Interface receives user data from the PeopleSoft database, it can map the data objects to the corresponding objects in the directory.

For PeopleSoft Directory Interface to map PeopleSoft information to your directory, it needs information about the directory hierarchical structure, or *directory information tree.*

Entries are made up of a *distinguished name* (DN) and attribute and value pairs. The distinguished name identifies an entry's position in the tree, and the attributes hold the data that make up the entry.

Available attributes for an object class entry are specified in the directory schema. You must load the schema into the Directory Interface before you can map PeopleSoft data to the directory.

PeopleSoft Directory Interface mapping tables map LDAP attributes to PeopleSoft messages. Each message contains selected information about a PeopleSoft record and its fields.

Note: Refer to PeopleSoft application documentation for information about specific messages delivered by PeopleSoft applications.

Understanding Data Synchronization

After you have loaded PeopleSoft data into your LDAP directory, you can synchronize the data. To do this, use one of the following options:

- PeopleSoft Business Interlinks.

PeopleSoft Business Interlinks updates the data in real time, so that your directory information is always synchronized with PeopleSoft data.

- LDAP Data Interchange Format (LDIF) files.

  You can load LDIF files as needed or defined by your system.

Note: The application server needs to be configured for receiving messages.

Delivered Business Interlinks

Oracle delivers the following business interlinks with PeopleSoft Directory Interface:

EO_DS_ADD Adds a new entry to the directory by creating a distinguished name and its corresponding attributes.

EO_DS_BIND Authenticates the information exchanged between the database and the directory.

EO_DS_DEL Deletes an entry from the directory.

EO_DS_MODDN Renames a directory entry. Changes its distinguished name by renaming the actual entry or changing its position in the directory entry.

EO_DS_MODIFY Changes the attributes of an entry.

EO_DS_SEARCH Searches for directory entries and their corresponding attributes.

Refer to *Enterprise PeopleTools 8.46 PeopleBook: PeopleSoft Business Interlinks* for more information on business interlinks.

Understanding Implementing PeopleSoft Directory Interface

Consider these PeopleSoft Directory Interface implementation guidelines for best results:

LDAP Searches Some LDAP searches may generate LDAP referrals to other servers participating in your directory. You must be able to ping by hostname all servers in the directory from the application server. If any server is unreachable by hostname from the application server, you can add a line for the server to the hosts. Your directory information tree must have user entries at the leaf level. This is required when an entry needs to be moved from one branch to another. The entry needs to be at the leaf level so that the system can read user attributes, one of which is the password file on the application server.

Entry Limit In the directory, configure the entry limit value to be larger than the number of rows that you expect will be returned. The default value is usually not sufficient.

Directory Tree

Your directory information tree must have user entries at the leaf level. This is required when an entry needs to be moved from one branch to another. The entry needs to be at the leaf level so that the system can read user attributes, one of which is the password.

Microsoft Active Directory

The following items apply to implementations that use Microsoft Active Directory:

- The registry key HKLM\System\CurrentControlSet \Services\NTDS\Parameters\Schema Update Allowed must be present and set to a nonzero DWORD on the Active Directory FSMO Role Owner.
- When creating structural object classes in Microsoft Active Directory, you need to specify containment. PsftJobs can be children of the following classes of objects only: builtinDomain, organizationalUnit, and domainDNS.
- You must add the server names in the Directory Setup component as they appear on the DNSHost Name attribute on the server entries under the CN=Sites entry.

Overview of Using PeopleSoft Directory Interface

This section briefly describes the steps needed to use PeopleSoft Directory Interface, including:

- Setting up in PeopleSoft Application Designer and PeopleSoft Integration Broker.
- Using the Directory Configurations component.

Setting Up in PeopleSoft Application Designer and PeopleSoft Integration Broker

Perform the following steps in PeopleSoft Application Designer and PeopleSoft Integration Broker.

Setting Up in PeopleSoft Application Designer

Access PeopleSoft Application Designer.

- Create authentication and user profile maps as needed.

  If you are going to authenticate users with the directory server, a PeopleSoft user profile is required —that is, a row in the PSOPRDEFN table where PeopleSoft user information is stored. In this context, you cache LDAP user information inside your PeopleSoft system. Properties that you specify in the Mandatory and Optional Properties pages of the Mappings component are the columns in PSOPRDEFN that the system populates with values from your directory server. PeopleSoft applications use this cache of user information, not your directory server. Whenever a transaction requires user information, the application refers to the local PSOPRDEFN table instead of querying the directory server.

- Add Signon PeopleCode.

  Directory authentication requires that Signon PeopleCode be enabled and configured with proper permissions. After a user signs onto the system and the Signon PeopleCode runs, the PeopleSoft system creates a row for the user in the user definition table by retrieving the LDAP information and creating a local cache. Signon PeopleCode maintains this row automatically and any changes made in the directory server are reproduced in the local cache. Using the Mappings component, set up mappings. To keep the data synchronized, you must map PeopleSoft data to the equivalent directory objects. PeopleSoft Directory Interface then associates the fields in the message with the attributes in the directory and updates the selected directory attributes with the field data from the message.

- Activate the DSCHNL channel.

  Open the message channel and select *Run.*

See *PeopleTools: PeopleSoft Application Designer Developer's Guide*

Setting Up in PeopleSoft Integration Broker

Access PeopleSoft Integration Broker.

- Activate a relevant node.

  This node should be the default local node.

- Define a service operation.

  Note that the service operations, and messages to include in the service operations, depend on the application. For example, in an human resources implementation, you might want to include messages such as Dept, Location, Person, and Job in a service operation, in addition to core messages such as DSMINPUT.

See *PeopleTools: PeopleSoft Integration Broker*

Using the Directory Configurations Component

Access Directory Configurations component (PSDSSETUP) from the browser menu.

- Using the Directory Configurations component, configure the directory.

  Enter appropriate connection information such as the server name (DNS or IP address) and the listening port number, the user DN, and associated password.

- Using the Schema Management page, select names of object classes and attribute types and then cache the schema.

- To keep the data synchronized, you must map PeopleSoft data to the equivalent directory objects.

  Set up mappings using the Mappings component. After this is completed, PeopleSoft Directory Interface associates the fields in the message to the attributes in the directory and updates the selected directory attributes with the field data from the message.

- Using the Membership Rules component, create rules and memberships, if desired.

- Load data in the directory.

- Set directory search criteria.

  Enter search parameters to query the directory and view the results.

Common Elements Used in The Directory Interface

Directory ID

Unique identifier for the directory.

Description

A brief description of the directory.

Directory Product

Select the directory product from the drop-down list box.

Default Connect DN

Displays the connect distinguished name associated with the directory ID that you selected. Use this ID to connect to the directory server.

Password

Password to access the directory.

LDAP Server

The name of the server where the directory resides.

Port

The LDAP server port associated with the LDAP server that you select.

SSL Port

The secure socket layer port.

Defining and Configuring the Directory

Use the Directory Configurations component (PSDSSETUP) to define and configure the directory connection. This section discusses how to:

- Configure the directory connection.
- Cache the schema.
- Delete the directory configuration.

Pages Used to Define and Configure the Directory

Page Name	***Definition Name***	***Navigation***	***Usage***
Directory Setup	DSDIRSETUP	Enterprise Components, Directory Interface, Definitions, Directory Configurations, Directory Setup	Enter values to configure the directory.

Page Name	***Definition Name***	***Navigation***	***Usage***
Additional Connect DN's	DSSERVERID	Enterprise Components, Directory Interface, Definitions, Directory Configurations, Additional Connect DN's	Add values for additional connect DNs.
Schema Management	DSEXTINSTALL	Enterprise Components, Directory Interface, Definitions, Directory Configurations, Schema Management	Manage schema, and apply PeopleSoft schema extensions.
Test Connectivity	DSSRCHRSLT	Enterprise Components, Directory Interface, Definitions, Directory Configurations, Test Connectivity	Test the directory connectivity.
Cache Schema	DSSCHEMACACHE	Enterprise Components, Directory Interface, Definitions, Schema Cache	Cache the schema.
Delete Directory	DSPURGEDIRID	Enterprise Components, Directory Interface, Definitions, Directory Deletions	Delete the directory configuration.

Configuring the Directory Connection

This section discusses how to:

- Set up the directory connection.
- Connect additional DN's.
- Manage the schema.

Directory Setup Page

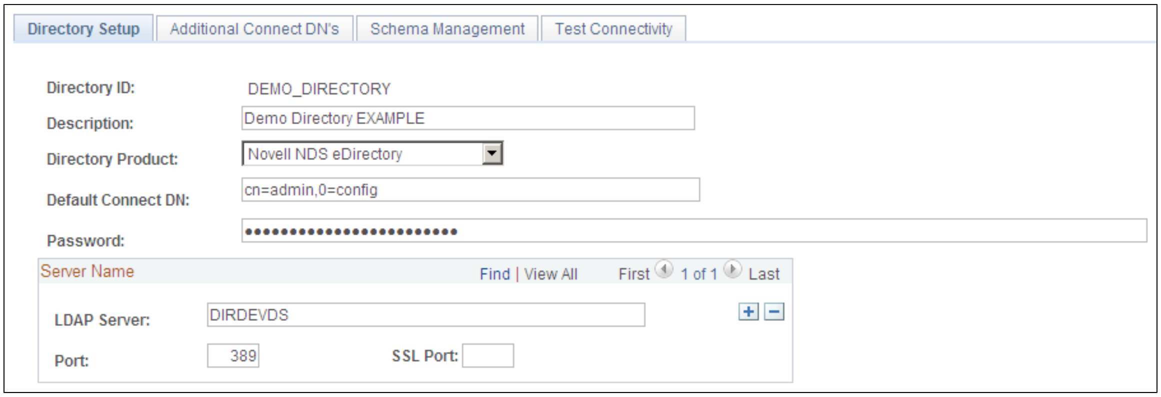
Use the Directory Setup page (DSDIRSETUP) to enter values to configure the directory.

Navigation

Enterprise Components, Directory Interface, Definitions, Directory Configurations, Directory Setup

Image: Directory Setup page

This example illustrates the fields and controls on the Directory Setup page. You can find definitions for the fields and controls later on this page.



Directory ID: DEMO_DIRECTORY

Description: Demo Directory EXAMPLE

Directory Product: Novell NDS eDirectory

Default Connect DN: cn=admin,0=config

Password: •••••••••••••••••••••••

LDAP Server: DIRDEVDS

Port: 389

SSL Port:

Additional Connect DN's Page

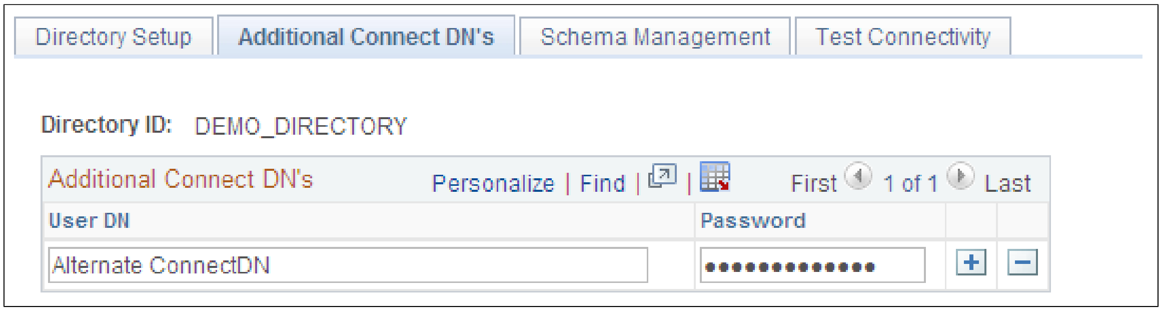
Use the Additional Connect DN's page (DSSERVERID) to add values for additional connect DNs.

Navigation

Enterprise Components, Directory Interface, Definitions, Directory Configurations, Additional Connect DN's

Image: Additional Connect DN's page

This example illustrates the fields and controls on the Additional Connect DN's page. You can find definitions for the fields and controls later on this page.



Directory ID: DEMO_DIRECTORY

User DN	Password
Alternate ConnectDN	•••••••••••••

Use this page to add values for additional connect DNs. Add more connect DNs and passwords, if needed.

Schema Management Page

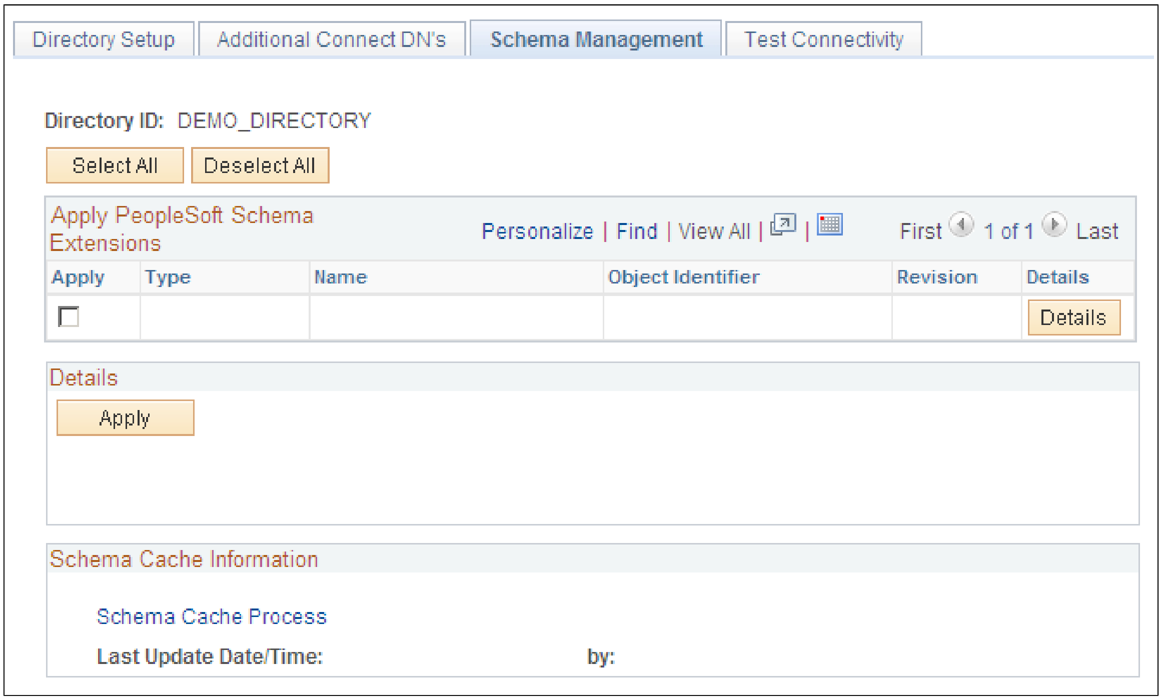
Use the Schema Management page (DSEXTINSTALL) to manage schema, and apply PeopleSoft schema extensions.

Navigation

Enterprise Components, Directory Interface, Definitions, Directory Configurations, Schema Management

Image: Schema Management page

This example illustrates the fields and controls on the Schema Management page. You can find definitions for the fields and controls later on this page.



Directory ID: DEMO_DIRECTORY

Apply PeopleSoft Schema Extensions

Apply	Type	Name	Object Identifier	Revision	Details
☐					Details

Details

Schema Cache Information

Last Update Date/Time: by:

Use this page to manage schema, and apply PeopleSoft schema extensions.

Activate the check boxes of those object classes or attribute types that you want applied to the cache schema.

Test Connectivity Page

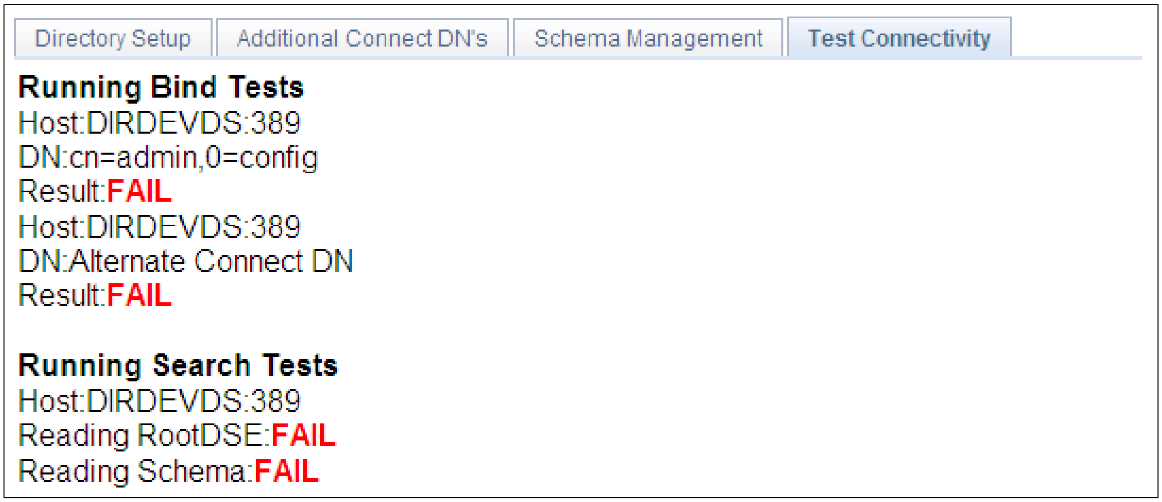
Use the Test Connectivity page (DSSRCHRSLT) to test the directory connectivity.

Navigation

Enterprise Components, Directory Interface, Definitions, Directory Configurations, Test Connectivity

Image: Test Connectivity page

This example illustrates the fields and controls on the Test Connectivity page. You can find definitions for the fields and controls later on this page.



Running Bind Tests
Host:DIRDEVDS:389
DN:cn=admin,0=config
Result:FAIL
Host:DIRDEVDS:389
DN:Alternate Connect DN
Result:FAIL

Running Search Tests
Host:DIRDEVDS:389
Reading RootDSE:FAIL
Reading Schema:FAIL

Use this page to test the directory connectivity.

When you access the Test Connectivity page, the connection test launches automatically. The results appear in the page. A successful test shows the message *SUCCESS* in green.

The preceding example shows the message *FAILED* in red, confirming that the connection test failed.

Verify that your directory server configuration details contain the correct values (correct server name, port, and so on).

Cache Schema Page

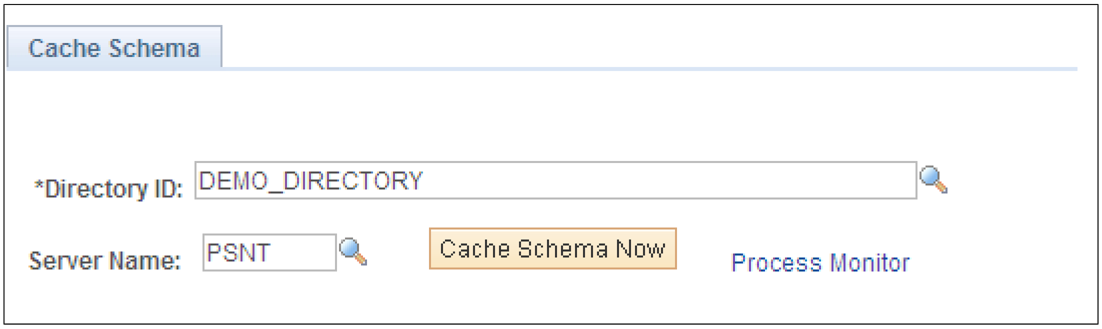
Use the Cache Schema page (DSSCHEMACACHE) to cache the schema.

Navigation

Enterprise Components, Directory Interface, Definitions, Schema Cache

Image: Cache Schema page

This example illustrates the fields and controls on the Cache Schema page. You can find definitions for the fields and controls later on this page.



Cache Schema

*Directory ID: DEMO_DIRECTORY

Server Name: PSNT

Enter the directory ID and server name of the schema to be cached and click the Cache Schema Now button.

Delete Directory Page

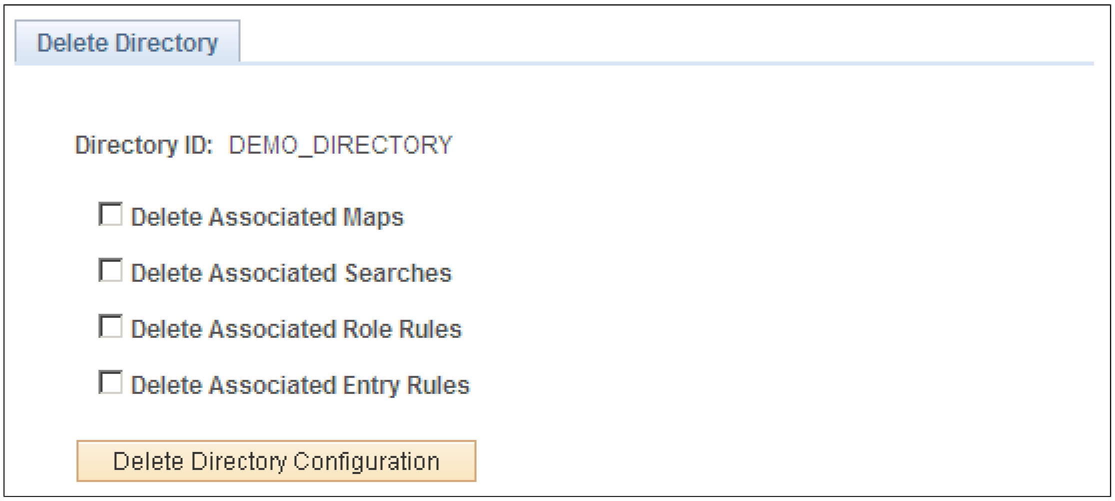
Use the Delete Directory page (DSPURGEDIRID) to delete the directory configuration.

Navigation

Enterprise Components, Directory Interface, Definitions, Directory Deletions

Image: Delete Directory page

This example illustrates the fields and controls on the Delete Directory page. You can find definitions for the fields and controls later on this page.



Delete Directory

Directory ID: DEMO_DIRECTORY

☐ Delete Associated Maps

☐ Delete Associated Searches

☐ Delete Associated Role Rules

☐ Delete Associated Entry Rules

Select the check boxes for the desired directory configuration deletions.

(Optional) Setting Up Directory Authentication

This section discusses how to:

- Use map authentication.
- View user properties.

For information about setting up authentication servers, user profile maps, and role membership rules, refer to the following documentation.

Pages Used to Set Up Directory Authentication

Page Name	***Definition Name***	***Navigation***	***Usage***
Authentication	DSSECMAPMAIN	Enterprise Components, Directory Interface, Mappings, Authentication	Create a mapping for the directory that the system relies on for authenticating users.
Mandatory User Properties	DSUSRPRFLMANMAP	Enterprise Components, Directory Interface, Mappings, User Profiles, Mandatory User Properties	Specify the attributes required for sign-in. You can have the system retrieve these mandatory values from the directory server, or you can enter default values.
Optional User Properties	DSUSRPRFLOPTMAP	Enterprise Components, Directory Interface, Mappings, User Profiles, Optional User Properties	Specify optional user properties to store in and retrieve from the directory. You can specify general, permission list, and workflow attributes. All these attributes appear in the User Profile component.

Authentication Page

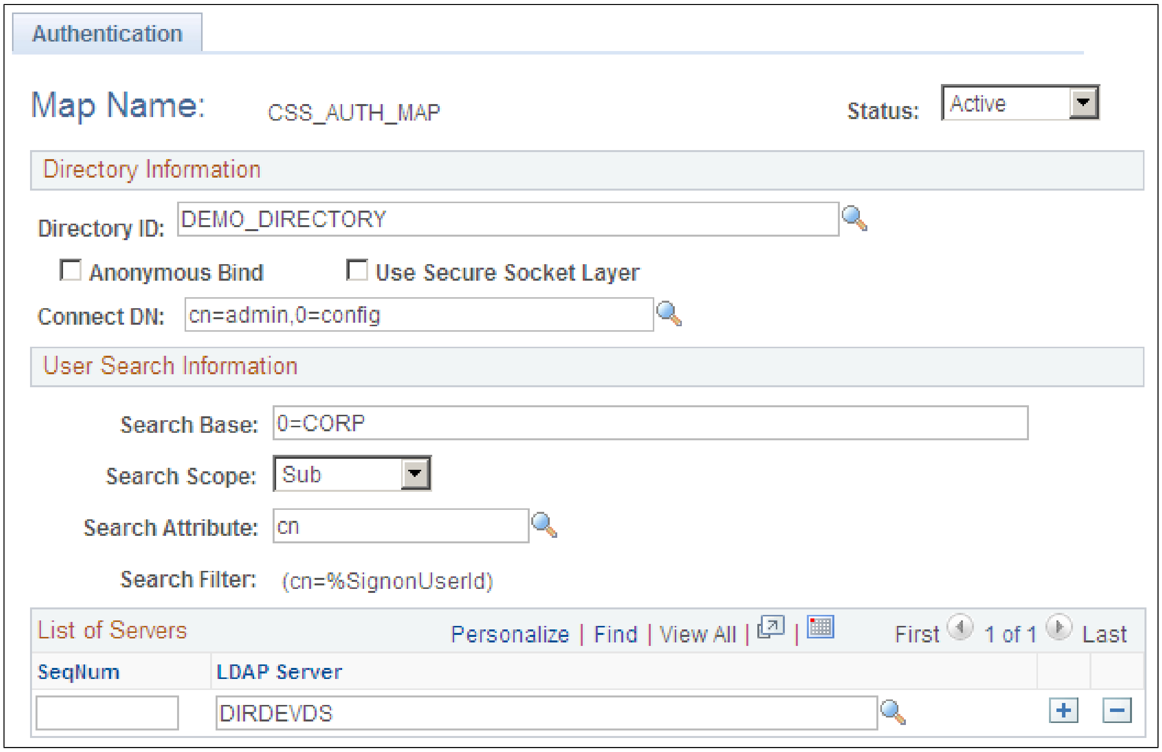
Use the Authentication page (DSSECMAPMAIN) to create a mapping for the directory that the system relies on for authenticating users.

Navigation

Enterprise Components, Directory Interface, Mappings, Authentication

Image: Authentication page

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



Map Name: CSS_AUTH_MAP

Status: Active

Directory Information

Directory ID: DEMO_DIRECTORY

□ Anonymous Bind □ Use Secure Socket Layer

Connect DN: cn=admin,0=config

User Search Information

Search Base: 0=CORP

Search Scope: Sub

Search Attribute: cn

Search Filter: (cn=%SignonUserId)

List of Servers

SeqNum	LDAP Server
	DIRDEVDS

Use this page to create a mapping for the directory that the system relies on for authenticating users.

Anonymous Bind

If directory data required for authentication and user profile maintenance is visible to an anonymous connection, you can select this check box.

Use Secure Socket Layer

Select if you are using SSL between the PeopleSoft system and the directory server.

Mandatory or Optional User Properties page

Use the Mandatory User Properties page (DSUSRPRFLMANMAP) to specify the attributes required for sign-in.

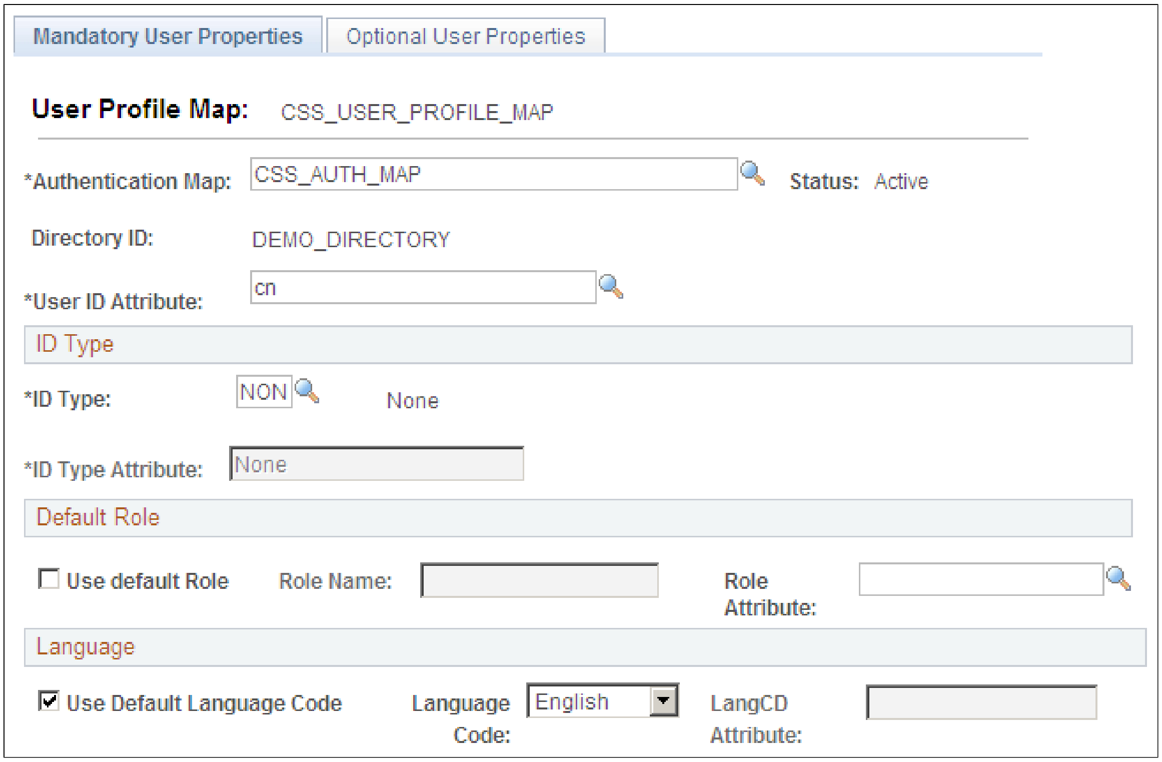
You can have the system retrieve these mandatory values from the directory server, or you can enter default values.

Navigation

Enterprise Components, Directory Interface, Mappings, User Profiles, Mandatory User Properties

Image: Mandatory User Properties page

This example illustrates the fields and controls on the Mandatory User Properties page. You can find definitions for the fields and controls later on this page.



User Profile Map: CSS_USER_PROFILE_MAP

*Authentication Map: CSS_AUTH_MAP

Status: Active

Directory ID: DEMO_DIRECTORY

*User ID Attribute: cn

ID Type

*ID Type: NON None

*ID Type Attribute: None

Default Role

□ Use default Role

Role Name:

Role Attribute:

Language

☑ Use Default Language Code

Language Code: English

LangCD Attribute:

Select Authentication Map and set check boxes and field values as needed.

Use the Optional User Properties page (DSUSRPRFLOPTMAP) to specify optional user properties to store in and retrieve from the directory.

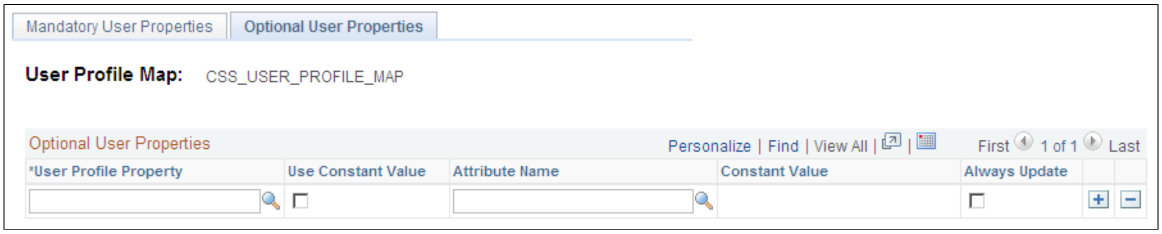
You can specify general, permission list, and workflow attributes. All these attributes appear in the User Profile component.

Navigation

Enterprise Components, Directory Interface, Mappings, User Profiles, Optional User Properties

Image: Optional User Properties page

This example illustrates the fields and controls on the Optional User Properties page. You can find definitions for the fields and controls later on this page.



User Profile Map: CSS_USER_PROFILE_MAP

Use the Mandatory User Properties page or the Optional User Properties page to specify the attributes required for sign-in. You can have the system retrieve these mandatory values from the directory server, or you can enter default values.

The default shows the Attribute Name field available. If you select the Use Constant Value check box, the Constant Value field becomes available instead.

Setting Up Mappings

This section provides an overview of mapping and discusses how to:

- Create mappings.
- Modify a distinguished name.
- Specify distinguished name details.
- Translate or perform functions with database values.
- Specify distinguished name defaults.
- Map PeopleSoft data to directory object class attributes.
- Locate delivered messages.

Pages Used to Set Up Mappings

Page Name	***Definition Name***	***Navigation***	***Usage***
Map Details	EO_DSMAP	Enterprise Components, Directory Interface, Mappings, Directory maps, Map Details	Set up a mapping and enter the data relationship details between PeopleSoft data and directory data.
Modify Connect DN - Directory Interface	EO_DSUSERDN	Click the Modify Connect DN button on the Map Details page.	Modify the Connect DN.

Page Name	***Definition Name***	***Navigation***	***Usage***
DN Details	EO_DSDN	Enterprise Components, Directory Interface, Mappings, Directory maps, DN Details	Set up the relationship between the data contained in the message that you selected on the Message Details page and the directory entry's distinguished name.
DN Defaults	EODS_DN_DEFAULTS	Enterprise Components, Directory Interface, Mappings, Directory maps, DN Defaults	Define a constant value or PeopleCode function that returns values that supply the blank values on the Directory Information Tree.
Attribute Details	EO_DSATTRIB	Enterprise Components, Directory Interface, Mappings, Directory maps, Attribute Details	Set up the relationship between the data in the message that you selected on the Message Details page and the directory object class attributes.

Understanding Mapping

You map PeopleSoft data to the equivalent directory objects to keep the data synchronized. PeopleSoft Directory Interface receives PeopleSoft data from messages contained within service operation that you publish whenever a business event associated with the messages identified in the Directory Mapping component. Each message contains information about records and the most recent data for the record fields. Using the mapping information that you set up, PeopleSoft Directory Interface associates the fields in the message with the attributes in the directory and then updates the selected directory attributes with the field data from the message. Additionally, you can define a constant value or a PeopleCode function that returns a value to supply data used in building temporary Directory Information Trees when not all data exists for an entry.

Map Details Page

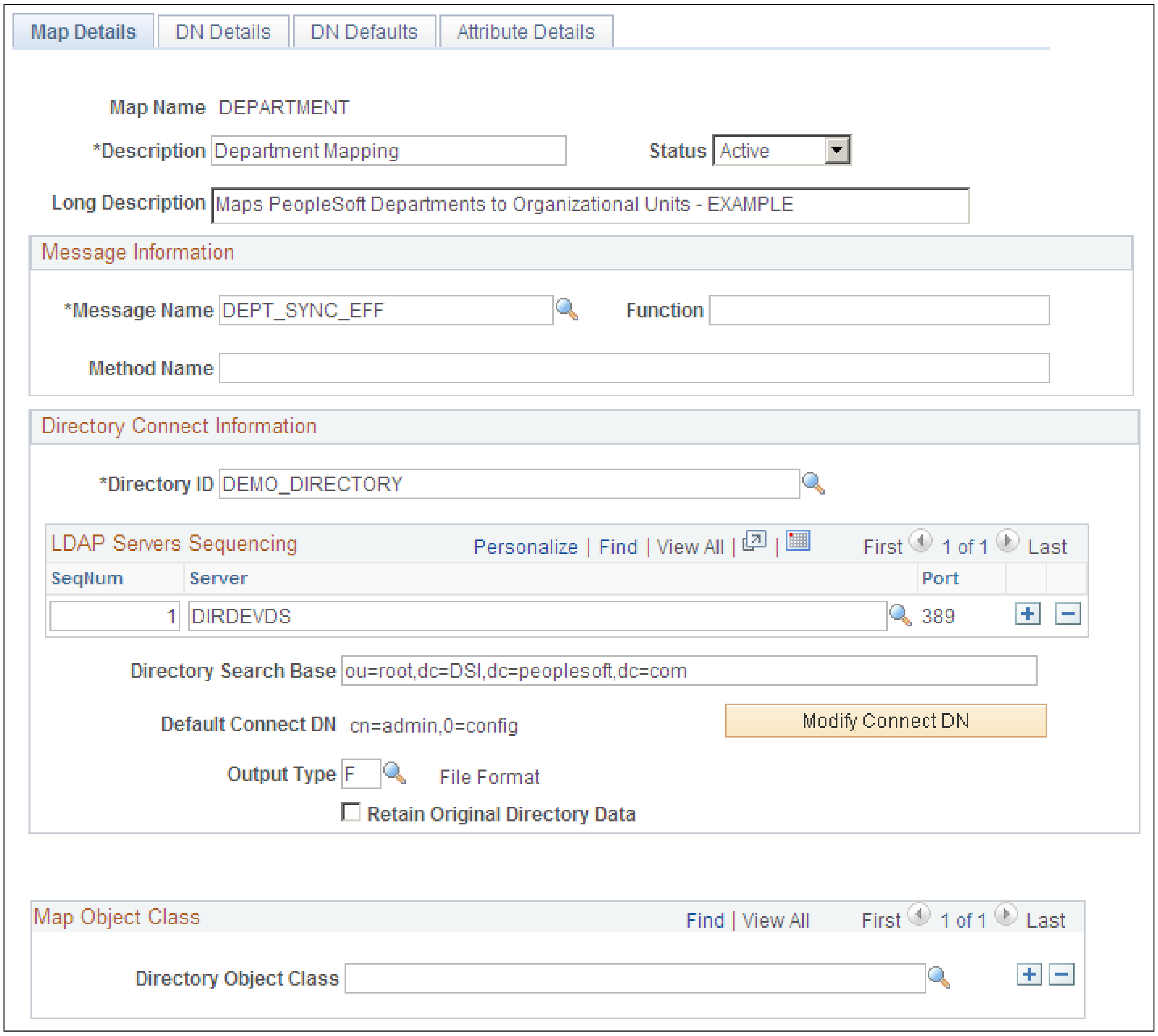
Use the Map Details page (EO_DSMAP) to set up a mapping and enter the data relationship details between PeopleSoft data and directory data.

Navigation

Enterprise Components, Directory Interface, Mappings, Directory maps, Map Details

Image: Map Details page

This example illustrates the fields and controls on the Map Details page. You can find definitions for the fields and controls later on this page.



Map Name DEPARTMENT

*Description Department Mapping

Status Active

Long Description Maps PeopleSoft Departments to Organizational Units - EXAMPLE

Message Information

*Message Name DEPT_SYNC_EFF

Function

Method Name

Directory Connect Information

*Directory ID DEMO_DIRECTORY

LDAP Servers Sequencing

SeqNum	Server	Port
1	DIRDEVDS	389

Directory Search Base ou=root,dc=DSI,dc=peoplesoft,dc=com

Default Connect DN cn=admin,0=config

Output Type F File Format

□ Retain Original Directory Data

Map Object Class

Directory Object Class

Use this page to set up mapping and enter the data relationship details between PeopleSoft data and directory data.

Status

Select the appropriate status from the following values.

- *Active:* The map is active and ready to be used.
- *Inactive:* The map is not ready to be used.
- *Remote:* The map is not used at this time, and may appear to be unavailable.

Message Information

Message Name

Select the message to associate with this mapping. The message contains the PeopleSoft records and fields that have the data that you want to associate with the attributes that make up the directory entry that you select in the Directory Connect Information group box. For example, if you select the output −DEPTID object class, select the department (DSDEPT_SYNC) message because it contains the fields relevant to the department object class.

Function

Enter the name of the PeopleCode function that you want to run using this message as an input parameter. The function can use any of the fields contained in the message to produce an output value for one or more of the fields that you map. This enables you to use a field in a function without mapping to it directly. For example, if you want the employee ID value sent to the directory to be a value combining the employee ID and the salary code, enter a function that produces that value. You then need to map only to the EmplID field to insert the derived employee ID in the directory.

Directory Connect Information

SeqNum (directory sequence number)

Indicate the order in which the server should be used when the system processes this mapping. If the first server is unavailable, the system attempts to access the other servers in sequence until it finds an available one. If you are using multiple servers, this enables you to distribute the load across servers.

Directory Search Base

Enter a directory search base. The search base is the entry in the directory information tree from which the system begins a search relating to this mapping. For example, if on the Attribute Details page you select to have a field value updated indirectly, PeopleSoft Directory Interface searches for and updates all instances of that field in entries from that point in the information tree down.

Modify Connect DN

Click to access the Modify Connect DN - Directory Interface page to modify the connect DN.

Output Type

Select the method that the system should use to send the mapped data to the directory data. Select *I* to send data to the directory directly through a business interlink. Select *F* to send data to an LDAP Data Interchange Format (LDIF) file to be manually updated in the directory.

Use the same output type for all your mappings to keep data consistent in the directory.

Retain Original Directory Data

When you move data in your directory using the PeopleSoft Directory Interface, the Directory Interface copies the data to the new location and then deletes the old version. Select this check box to preserve the original data. You can select this check box at a later date provided that you do it before the data move.

Note: Select this check box if your directory contains binary data. Move the binary data with your directory administrative tool.

Map Object Class

Directory Object Class

Select one or more directory object classes. The object classes that you select determine the attributes that you can map to PeopleSoft data.

Modify Connect DN - Directory Interface Page

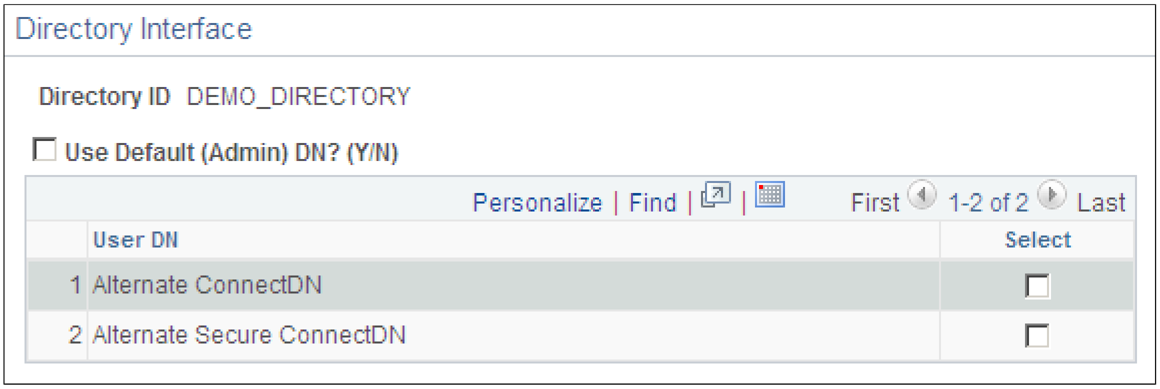
Use the Modify Connect DN - Directory Interface page (EO_DSUSERDN) to modify the Connect DN.

Navigation

Click the Modify Connect DN button on the Map Details page.

Image: Modify Connect DN - Directory Interface page

This example illustrates the fields and controls on the Modify Connect DN - Directory Interface page. You can find definitions for the fields and controls later on this page.



Directory Interface

Directory ID DEMO_DIRECTORY

□ Use Default (Admin) DN? (Y/N)

	User DN	Select
1	Alternate ConnectDN	□
2	Alternate Secure ConnectDN	□

Use this page to modify the connect DN.

Use Default (Admin) DN? (Y/N) (use default [administrative] distinguished name)

Select to use the default connect distinguished name value that you set up in PeopleTools.

User DN (user distinguished name)

Displays the alternative IDs that you can use to connect to the specified directory ID. You can use a user ID (and password) other then the default one listed on the Directory Setup page in PeopleTools. Because the default user ID is most likely an administrative ID, you can set up a more secure user ID for the scope of the mapping.

DN Details Page

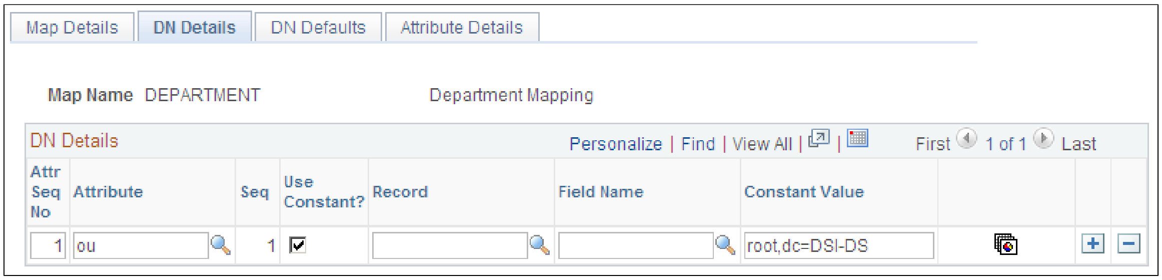
Use the DN Details page (EO_DSDN) to set up the relationship between the data contained in the message that you selected on the Message Details page and the directory entry’s distinguished name.

Navigation

Enterprise Components, Directory Interface, Mappings, Directory maps, DN Details

Image: DN Details page

This example illustrates the fields and controls on the DN Details page. You can find definitions for the fields and controls later on this page.



Map Name DEPARTMENT Department Mapping

DN Details

Attr Seq No	Attribute	Seq	Use Constant?	Record	Field Name	Constant Value
1	ou	1	☑			root,dc=DSI-DS

Use this page to define a constant value or PeopleCode function that returns values that populate the blank values on the Directory Information Tree.

Associate the data contained in the message that you selected on the Map Details page with the entry's distinguished name.

DN Details

Attr Seq No (attribute sequence number)

The system assigns an attribute sequence number to the attributes. Some directory attribute values consist of multiple values. The attribute sequence number distinguishes between the different attribute values and indicates to PeopleSoft Directory Interface the order in which the PeopleSoft values and constant values should be assigned to the attribute.

Attribute

Select the directory attributes associated with the mapping’s distinguished name. For example, for the Department entry, map the o – Corporation first, the l – location second, and then the ou – Department attribute.

Seq (sequence)

Enter the sequence number of the directory attribute. The directory builds the entry’s distinguished name using the attributes in sequential order.

Use Constant and Constant Value

Select to use the constant value that you enter in the Constant Value field to supply this attribute instead of a PeopleSoft field value.

Record and Field Name

Select the name of the record that contains the PeopleSoft field and the PeopleSoft field containing the value to assign to this attribute.

Click to access the DN Attribute Function - Directory Interface page and translate database values or instruct the system to perform functions with database values.

Note: Use this page when constructing distinguished names across active directory multiple domains.

Example Entry

An entry's distinguished name is built by applying the attributes in a sequential order. The order for the department entry example would be constructed using the data in the following table:

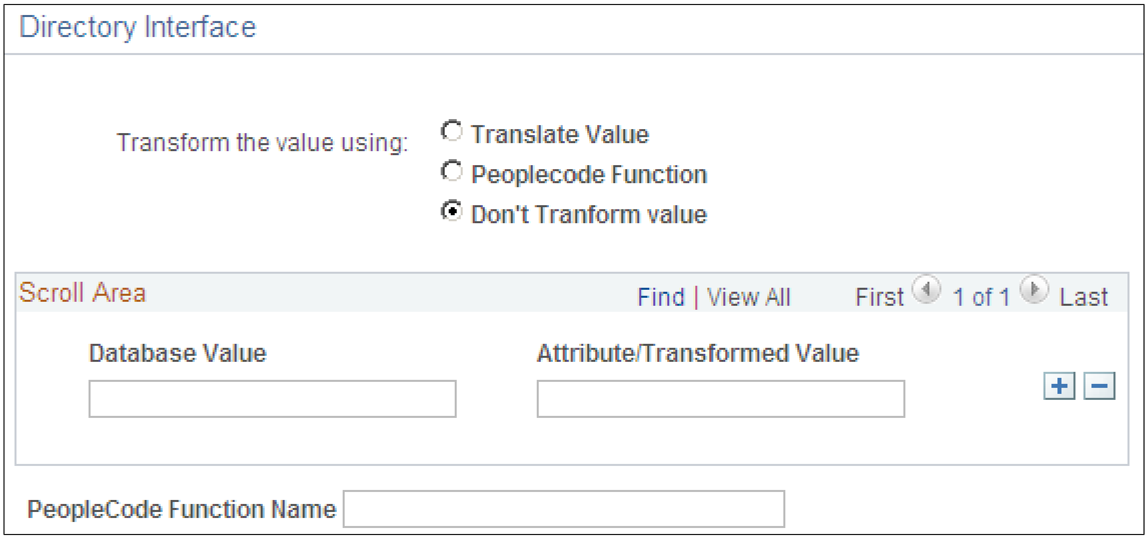
Sequence Number	***Directory Attribute***	***Attribute Sequence Number***	***Use Constant Value***	***Record (Table) Name***	***Field Name***	***Constant Value***
1	o	1	Yes			Corp
2	l	1	No	DEPT_TBL	LOCATION	NA
3	ou	1	No	DEPT_TBL	DEPTID	NA

Translating or Performing Functions with Database Values

Access the DN Attribute Function - Directory Interface page (Enterprise Components, Directory Interface, Mappings, Directory maps, Attribute Details).

Image: DN Attribute Function - Directory Interface page

This example illustrates the fields and controls on the DN Attribute Function - Directory Interface page. You can find definitions for the fields and controls later on this page.



Directory Interface

Transform the value using: ○ Translate Value ○ Peoplecode Function ● Don't Tranform value

Use this page to set up the relationship between the data in the message that you selected on the Message Details page and the directory object class attributes.

Translate Value

Select to replace the database value with the Distinguished Name field value for the selected attribute.

PeopleCode Function

Select to use the selected database object value as a parameter in a PeopleCode function. The system uses the resulting value as the attribute's distinguished name.

Don’t Transform value

Select to instruct the system to keep the database value as is. This option is the default value for this field.

Database Value

Enter the database value that you want the system to replace. For example, every time the database value *Vancouver* appears in the Location attribute, the system replaces it with the distinguished name *Van.*

This field is available only when you select Translate Value as the transformation option.

Distinguished Name

Enter the distinguished name value to replace the database value.

This field is available only when you select Translate Value as the transformation option.

PeopleCode Function Name

Enter the PeopleCode function that the system should use to calculate the distinguished name for the selected attribute.

This field is available only when you select PeopleCode Function as the transformation option.

Setting Up PeopleCode Attribute-Level Functions

When the mapping function accesses the values in the selected field, the field value is passed into a PeopleCode function as a parameter and the output is assigned to the attribute in the directory.

Before you can enter a function on this page in the PeopleCode Function Name field, you must set up the function in the FUNCLIB_EO_DS.DSDYNFUNC FieldFormula.

To create a function:

1. Open the FUNCLIB_EO_DS.DSDYNFUNC FieldFormula.
2. Add a section in DSDynamicAttrFunc.
3. In the evaluate statement, add the following section for each function that you want to add (*FuncX* is equal to your function name):

   ```
   When = 'FuncX'
      FuncX(&AttrIn, &AttrRT);
      Break;
   ```

4. Define a DSDynamicAttrFunc PeopleCode function.

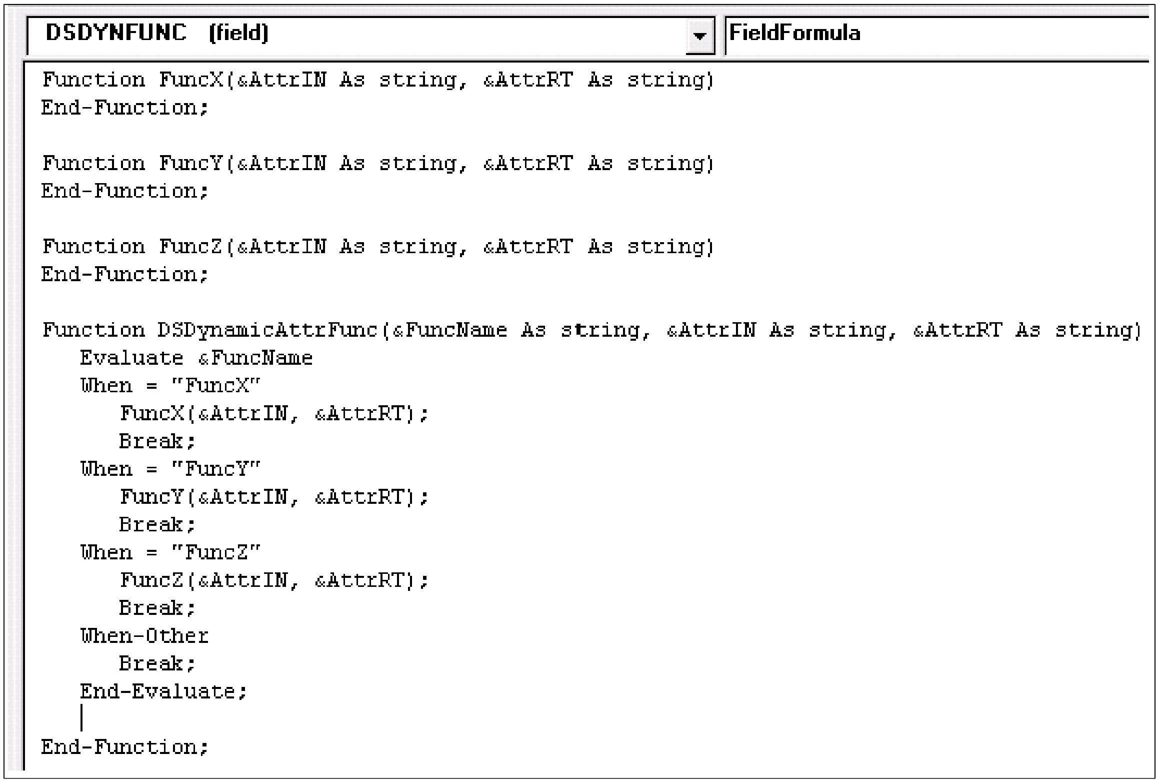
   The parameter list must contain two parameters, an attribute type string input and an attribute type string output.

PeopleCode Function Example

The following example displays the setup for functions FuncX, FuncY, and FuncZ.

Image: Setup functions on the FUNCLIB_EO_DS.DSDYNFUNC FieldFormula

This example illustrates the fields and controls on the Setup functions on the FUNCLIB_EO_DS.DSDYNFUNC FieldFormula. You can find definitions for the fields and controls later on this page.



```
Function FuncX(&AttrIN As string, &AttrRT As string)
End-Function;

Function FuncY(&AttrIN As string, &AttrRT As string)
End-Function;

Function FuncZ(&AttrIN As string, &AttrRT As string)
End-Function;

Function DSDynamicAttrFunc(&FuncName As string, &AttrIN As string, &AttrRT As string)
   Evaluate &FuncName
   When = "FuncX"
      FuncX(&AttrIN, &AttrRT);
      Break;
   When = "FuncY"
      FuncY(&AttrIN, &AttrRT);
      Break;
   When = "FuncZ"
      FuncZ(&AttrIN, &AttrRT);
      Break;
   When-Other
      Break;
   End-Evaluate;

End-Function;
```

DN Defaults Page

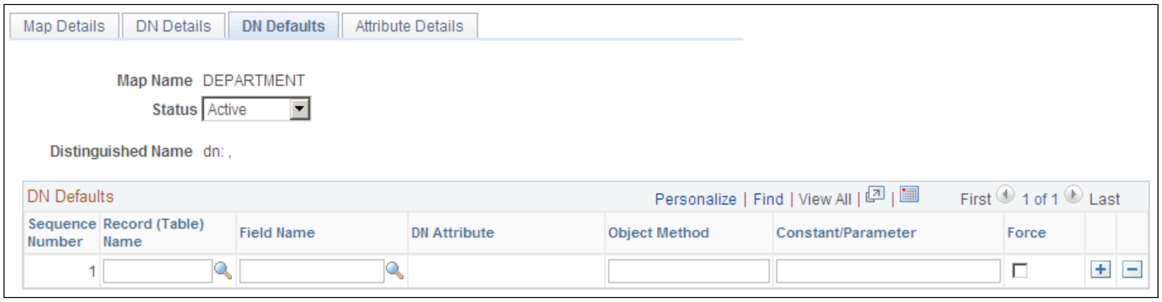
Use the DN Defaults page (EODS_DN_DEFAULTS) to define a constant value or PeopleCode function that returns values that supply the blank values on the Directory Information Tree.

Navigation

Enterprise Components, Directory Interface, Mappings, Directory maps, DN Defaults

Image: DN Defaults page

This example illustrates the fields and controls on the DN Defaults page. You can find definitions for the fields and controls later on this page.



Map Name DEPARTMENT

Status Active

Distinguished Name dn: ,

DN Defaults

Sequence Number	Record (Table) Name	Field Name	DN Attribute	Object Method	Constant/Parameter	Force
1						□

Use this page to define a constant value or PeopleCode function that returns values that populate the blank values on the Directory Information Tree.

This page enables you to define defaults for any Record.Field value that is left blank in the data rowset of the message that is used to populate the map. For example, you can enter defaults to enter the blanks left by the lack of Department/Location data for the Persons of Interest constructed by the PeopleSoft Directory Interface.

In the preceding sample page, the value for JOB.DEPTID is by default a method called HCDI_SERVICES:HCDIUtilities.DeptID. This method returns a DeptID constant.

Note: The syntax for the method needs to be fully qualified using the following format: Package_Name:App_Class_Name.Method_Name.

Seq (sequence number) Displays the sequence number for this attribute.

Record (Table) Name Select the record name for the value. This field is required.

Field Name Select the field name for the value. This field is required.

DN Attribute The name of the distinguished name attribute.

Object Method Enter the object method you are using to supply the value, if applicable. Leave this field blank if you are using a constant or parameter to supply the value.

Constant/Parameter Enter the values for the constant or the parameter, if applicable. Leave this field blank if you are using an object method to supply the value.

Force Select to overwrite the Record.Field values at runtime, even if the values exist.

Attribute Details Page

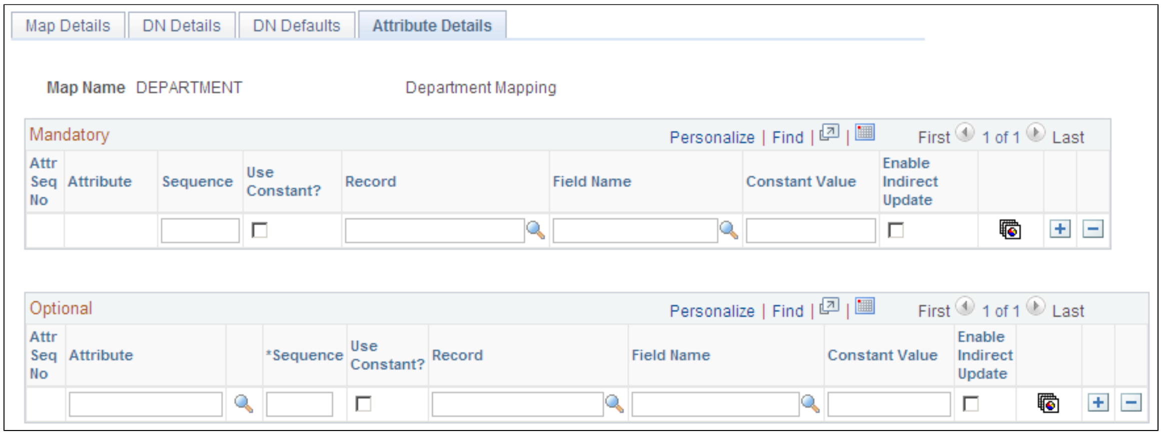
Use the Attribute Details page (EO_DSATTRIB) to set up the relationship between the data in the message that you selected on the Message Details page and the directory object class attributes.

Navigation

Enterprise Components, Directory Interface, Mappings, Directory maps, Attribute Details

Image: Attribute Details page

This example illustrates the fields and controls on the Attribute Details page. You can find definitions for the fields and controls later on this page.



Map Name DEPARTMENT Department Mapping

Mandatory

Attr Seq No	Attribute	Sequence	Use Constant?	Record	Field Name	Constant Value	Enable Indirect Update
			□				□

Optional

Attr Seq No	Attribute	*Sequence	Use Constant?	Record	Field Name	Constant Value	Enable Indirect Update
			□				□

Use this page to set up the relationship between the data in the message that you selected on the Message Details page and the directory object class attributes.

On this page, associate the fields contained in the message that you selected on the Map Details page with the attributes that provide more detail about an entry. Some attributes are mandatory (an object class's mandatory attributes are defined in the directory schema) and must be mapped to either a constant value or record or field. For the department example, you would map PeopleSoft records and fields to the mandatory attributes (such as DeptID), and you could add additional attributes that would give you more information about the object class, such as description.

Note: The system does not update related-display field values unless the source field is also mapped. If the source field is not mapped, the audit process still indicates and enables you to update any discrepancies. For example, when you map to an employee's job code, the directory entry also includes the job code description. If you change the job code description on the Job Code component, the system updates the related-display description field on the employee's Job Data page, but it does not update to the directory, because it is not included in the mapping.

Warning! The fields that you map to mandatory attributes must contain data or the mapping will fail. You can guarantee that data will be in the fields by mapping mandatory attributes to required fields.

Attr Seq No (attribute sequence number)

Displays the attribute sequence number assigned to this attribute.

Attribute

In the Mandatory scroll area, the system displays the mandatory attributes for this object class.

In the Optional scroll area, select optional attributes.

Seq (sequence number)

Enter a sequence number for this attribute. Some directory attribute values are made up of multiple values. The attribute sequence number distinguishes between the different attribute values and indicates to PeopleSoft Directory Interface the order in which the PeopleSoft values and constant values should be assigned to the attribute.

Ind Upd (indirect update)

Select if the field that you selected is used as an attribute in the directory outside of this mapping and you want it to be updated when this field is updated. The system updates attributes only in entries at lower levels on the directory information tree than this entry.

Locating Delivered Messages

Your PeopleSoft application that supports the PeopleSoft Directory Interface delivers a set of messages to be used to share information with your directory service.

Note: If you have upgraded from a PeopleTools 8.47 or earlier release, the upgrade program creates service operations for these messages. The service operation names and message names are the same.

For information about this delivered data and how it works in conjunction with the PeopleSoft Directory Interface, see your PeopleSoft application documentation.

(Optional) Setting Up Entry Membership Rules

This section discusses how to:

- Create entry definitions.
- Specify entry membership rules.

Entry membership rules enable you to modify a directory entry, such as a group, based on criteria stored in the PeopleSoft database. This feature provides a method to match any type of directory entry to rules that are meaningful in PeopleSoft applications. You can use membership rules to create any type of logical grouping in the directory. The groupings are not restricted to security purposes.

Pages Used to Set Up Entry Membership Rules

Page Name	***Definition Name***	***Navigation***	***Usage***
Entry Definition	EO_DSCONTAINERDEFN	Enterprise Components, Directory Interface, Membership Rules, Entry Rules, Entry Definition	Create a directory entry definition.

Page Name	***Definition Name***	***Navigation***	***Usage***
Entry Membership Rules	EO_DSSECRULES	Enterprise Components, Directory Interface, Membership Rules, Entry Rules, Entry Membership Rules	Establish entry membership rules.

Entry Definition Page

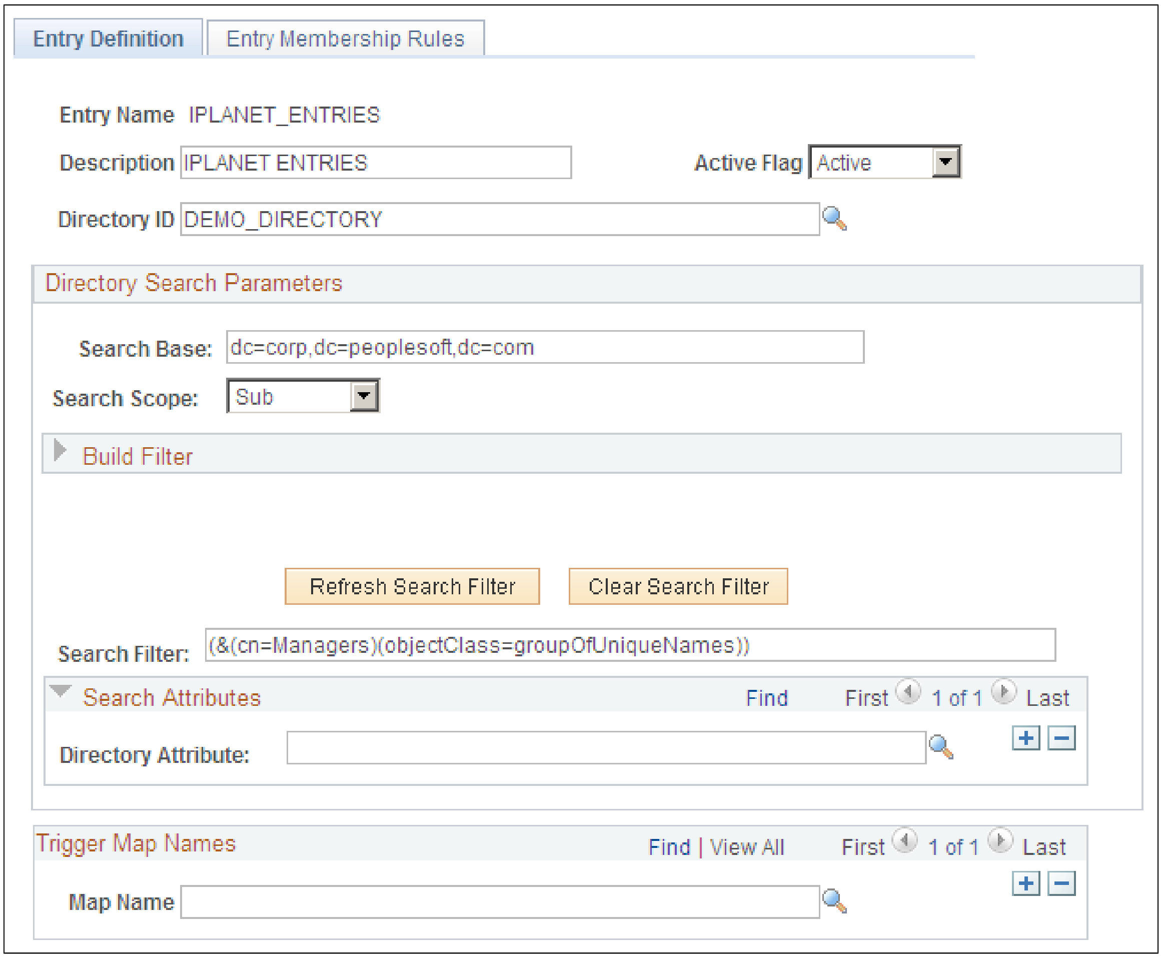
Use the Entry Definition page (EO_DSCONTAINERDEFN) to create a directory entry definition.

Navigation

Enterprise Components, Directory Interface, Membership Rules, Entry Rules, Entry Definition

Image: Entry Definition page

This example illustrates the fields and controls on the Entry Definition page. You can find definitions for the fields and controls later on this page.



Entry Name IPLANET_ENTRIES

Description IPLANET ENTRIES

Active Flag Active

Directory ID DEMO_DIRECTORY

Search Base: dc=corp,dc=peoplesoft,dc=com

Search Scope: Sub

Search Filter: (&(cn=Managers)(objectClass=groupOfUniqueNames))

Directory Attribute:

Map Name

Use this page to create a directory entry definition.

Entry Name Displays the entry name that you entered on the search page. The system uses this value for the entry name throughout the application, so it must be the name of an existing entry in the external directory. The PeopleSoft system assumes that the name is unique in the directory.

Active Flag

Select to activate rules. Rules that are not active do not run.

Directory Search Parameters

Search Base

Enter the distinguished name of the base under which this entry will be located in the directory. The application performs an LDAP search to retrieve the distinguished name of the entry using this field as the base.

Search Scope

Select from:

Base: The query searches only the value in the Search Base field.

One: The query searches only the entries one level down from the value in the Search Base field.

Sub: The query searches the value in the Search Base field and all entries beneath it.

Build Filter

( )

Select the check boxes below the parentheses to group expressions. You can group more than one line together using the check box on the left for the first line and the check box on the right for the last line.

Attribute

Enter the name of the attribute that will store the members of the entry in the external directory. It is typically set to *member,* but the attribute name could be anything that you choose.

Operation

Assign an operator to your rule such as <, <=, <>, =, >, or >=.

Value

Assign a value to the attribute in your rule.

And/Or

To add another line to your rule, select *AND* or *OR* depending on your rule logic. Select *END* to signify the end of the search. Select *NONE* if you are not using this kind of filter.

Refresh

After you make changes using the Build Filter options, click this button to update the Search Filter edit box to reflect the changes.

Clear LDAP Filter

Click to delete all values from the Search Filter edit box and the Build Filter selections.

Search Filter

Displays the filter that the system applies to the search for the distinguished name of the defined entry. This field typically displays the directory object class of the entry in the form “objectclass = GroupOfUniqueNames”, for example. This indicates what type of entry to search. To retrieve the correct entry distinguished names, the system adds the name of the entry to the search filter at runtime. The name retrieved by the LDAP search using this filter is tied to the rules defined in the Entry Membership Rules page. When these rules run, the employee that the system is currently processing is either added to or deleted from the distinguished name retrieved by the search.

Search Attributes

Directory Attribute

Select the attribute of the entry being defined that will contain all the members of this entry. This attribute must be valid for the current entry in the directory. The employees that satisfy the entry membership rules of this entry are added under this entry as a new value of this attribute. Because of this, as many attribute values may exist as employees satisfying the entry membership rules. If this field is left blank, the application uses *member* as a default attribute name.

Trigger Message Names

Map Names

Select the names of the maps to associate with the entry definition. Besides being a security feature, this also improves performance at runtime, because only applicable rules are evaluated.

Note: Run the directory audit if an entry rule has changed or if you want to initialize the directory entries.

Entry Membership Rules Page

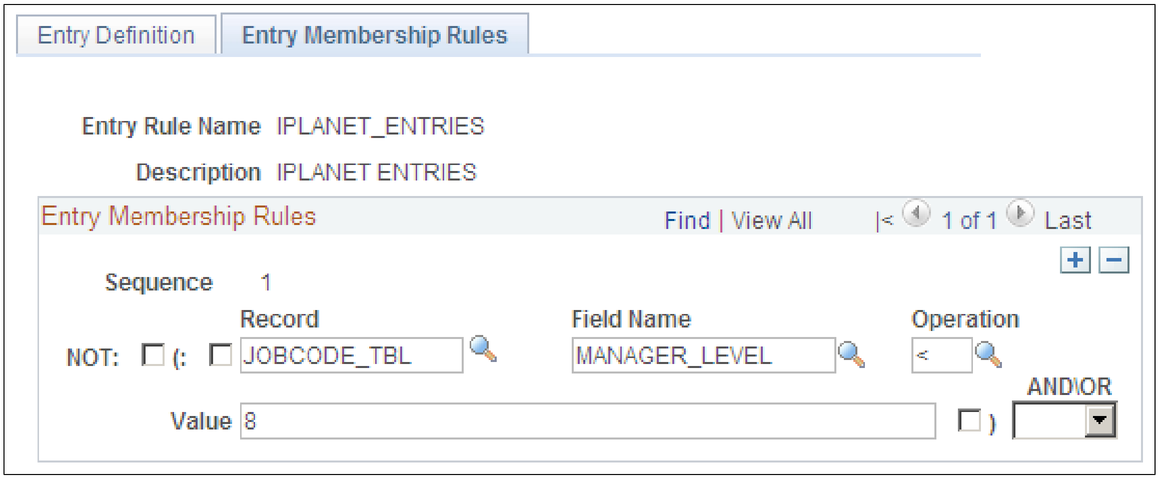
Use the Entry Membership Rules page (EO_DSSECRULES) to establish entry membership rules.

Navigation

Enterprise Components, Directory Interface, Membership Rules, Entry Rules, Entry Membership Rules

Image: Entry Membership Rules page

This example illustrates the fields and controls on the Entry Membership Rules page. You can find definitions for the fields and controls later on this page.



Entry Rule Name IPLANET_ENTRIES

Description IPLANET ENTRIES

Sequence 1

NOT: □ (: □ Record JOBCODE_TBL Field Name MANAGER_LEVEL Operation <

Value 8 □) AND\OR

Use this page to establish entry membership rules.

Entry Membership Rules

Sequence

Displays the sequence of a rule within a rule set. The sequence becomes significant when you enter more than one rule.

NOT

Select to negate the rule that you enter. This is similar to using the symbol *!* to reverse the truth value of an operand.

( )

Select the check boxes to add parentheses around your rule. You can group more than one line together using the check box on the left for the first line and the check box on the right for the last line.

Record and Field Name

Enter the name of the PeopleSoft record and field containing the information to be tested.

Operation

Enter the appropriate operator, such as: < , <= , <> , = , > , or >=.

Value

Enter the value on which the employee's data needs to be tested. This can be any value of the same type as the field used in the rule, such as String, number, date, and so on.

AND/OR

To add another line to your rule, select *AND* or *OR* depending on your rule logic. Select *END* to signify the end of the search. Select *NONE* if you are not using this kind of filter.

The entry rules are logical expressions that can be either true or false. They are composed of filters on database objects associated by logical operators. Rules have the following form:

```
[NOT] [ ( ]  Record . Field   operatorConstant [ ) ] [AND/OR]
```

The symbols between square brackets are optional. The operator can be <, <=, <>, =, >, or >=. A rule set is composed of single rules joined by AND or OR Boolean operators if necessary. The following example shows a series of single rules joined to make one compound rule.

```
( JOB.LOCATION = ‘KC004’ AND [1]
JOB.COMPRATE > 15000 ) OR [2]
NOT JOB.DEPTID = ‘GBIY004’[3]
```

Note: No limits are imposed on the number of rules used within a rule set.

Loading Data into the Directory

This section provides an overview of how to load the directory and discusses how to load the directory with PeopleSoft data.

Understanding Directory Load Behavior

Use the Directory Load process when no existing data is in the directory. The process overwrites any data in the directory.

If you have data in your directory, use the Directory Audit process instead of the Directory Load process. The audit process compares the PeopleSoft data to your existing directory data and enables you to review and resolve any possible conflicts.

Note: For PeopleSoft Human Capital Management (HCM) customers only, an alternative process named DSMAPINPUT FullSync is available that you can use in place of the Directory Load process. This new process does not replace the Directory Load process; it is provided as an alternative to load the data if performance becomes an issue.

See *PeopleSoft HCM: Application Fundamentals*

Directory Load Page

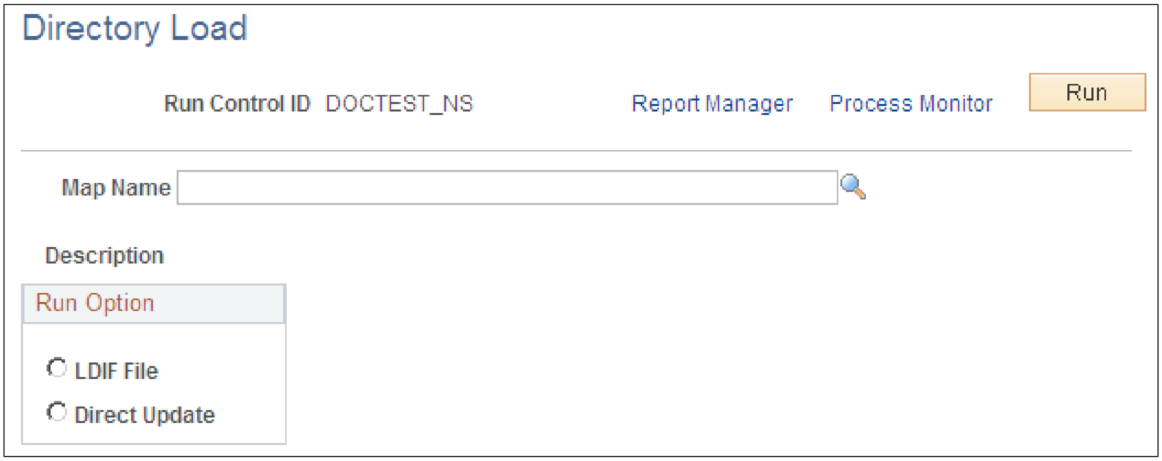
Use the Directory Load page (EO_RUNCTL_DS_LOAD) to load directories with PeopleSoft data.

Navigation

Enterprise Components, Directory Interface, Load Directory

Image: Directory Load page

This example illustrates the fields and controls on the Directory Load page. You can find definitions for the fields and controls later on this page.



Directory Load

Run Control ID DOCTEST_NS

Map Name

Description

Run Option

○ LDIF File

○ Direct Update

Use this page to run the Directory Load process.

LDIF File	Select to have the process send the data to an LDIF file for you to load in the directory.
Direct Update	Select to have the process directly update the directory.
Run	Click to run the process using PeopleSoft Process Scheduler.

Chapter 3

Reviewing Directory Data and Generating Reports

Reviewing LDAP Directory Data

This section discusses how to:

- Run a directory audit.
- Run a directory search.

Pages Used to Review LDAP Directory Data

Page Name	***Definition Name***	***Navigation***	***Usage***
Directory Audit	EO_RUNCTL_DS_AUDIT	Enterprise Components, Directory Interface, Run Directory Audit	Run the Directory Audit process.
Directory Search	EO_DSSRCHDIRECTORY	Enterprise Components, Directory Interface, Search Directory	Define search parameters to query the directory. The page saves the search parameters for future use.

Directory Audit Page

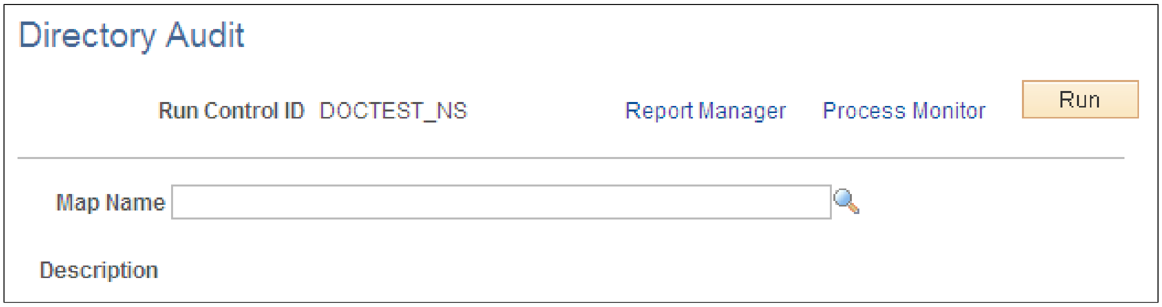
Use the Directory Audit page (EO_RUNCTL_DS_AUDIT) to run the Directory Audit process.

Navigation

Enterprise Components, Directory Interface, Run Directory Audit

Image: Directory Audit page

This example illustrates the fields and controls on the Directory Audit page. You can find definitions for the fields and controls later on this page.


Directory Audit

Run Control ID DOCTEST_NS

Map Name

Description

Use this page to run the process to ensure that your directory database has the same data as your PeopleSoft database.

The Directory Audit process compares the data in the directory to the data in the PeopleSoft database identified in the selected map and creates an LDAP Data Interchange Format (LDIF) file containing any discrepancies using the PeopleSoft system as the authority. You can then use the LDIF file to update the directory.

Note: For PeopleSoft HCM customers only, an alternative process named DSMAPINPUT FullSync Audit is available that you can use in place of the Directory Audit process. This new process does not replace the Directory Audit process; it is provided as an alternative to audit the data if performance becomes an issue.

See *PeopleSoft HCM: Application Fundamentals*

Map Name Select the name of the map that the audit should be run against.

Directory Search Page

Use the Directory Search page (EO_DSSRCHDIRECTORY) to define search parameters to query the directory.

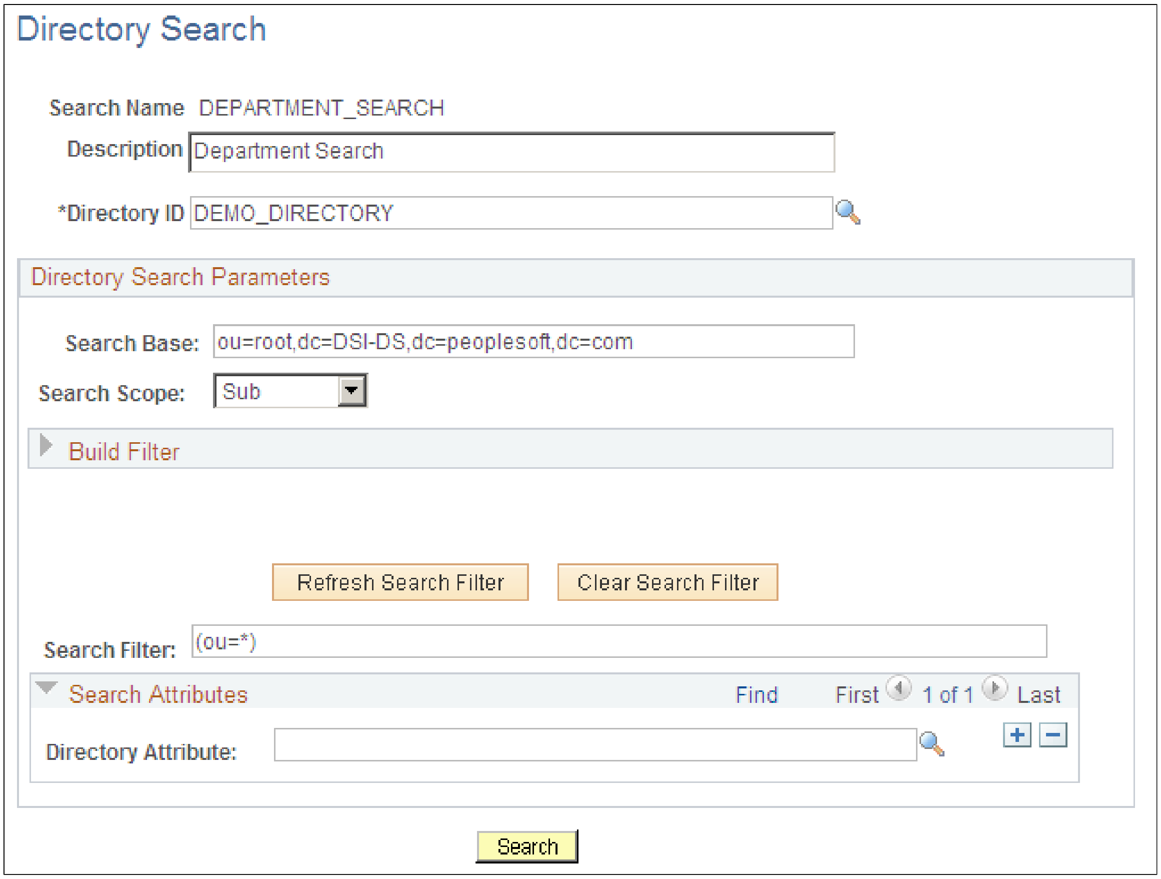
The page saves the search parameters for future use.

Navigation

Enterprise Components, Directory Interface, Search Directory

Image: Directory Search page

This example illustrates the fields and controls on the Directory Search page. You can find definitions for the fields and controls later on this page.



Search Name DEPARTMENT_SEARCH

Description Department Search

*Directory ID DEMO_DIRECTORY

Search Base: ou=root,dc=DSI-DS,dc=peoplesoft,dc=com

Search Scope: Sub

Search Filter: (ou=*)

Directory Attribute:

Use this page to define search parameters to query the directory and view the results. Search results are displayed on the Search Results page as they appear in the directory.

Search Name

Enter the search name. The system saves the search parameters that you enter on this page and stores them under this name for future use.

Directory Search Parameters

Search Base

Select the directory entry that is the search base for this search. The search base is the entry in the directory information tree at which the search begins querying.

Search Scope

Select from:

Base: The query searches only the value in the Search Base field.

One: The query searches only the entries one level down from the value in the Search Base field.

Sub: The query searches the value in the Search Base field and all entries beneath it.

Build Filter

Use the fields in the Build Filter group box to create an attribute-specific filter. For example, if you want data on a single person, enter the attribute name *Person,* the operation =*,* and the person's name in the Value field. You can construct multiple filters.

( ) Use parentheses to separate filter statements.

Attribute Name Enter the name of the attribute data to which you want to filter.

Operation Select an operator to determine how to filter the data.

Value Select a value to compare against when filtering.

And/Or Select from:

- *And:* A search must meet the criteria of multiple statements.
- *END:* This value marks the end of a query. This value applies parentheses appropriately.
- *NoOp:* No operator is used.
- *OR:* A search must meet the criteria of one of multiple statements.

Search Filter You can narrow the search (for example, instruct the system to search for all attributes but one) by entering a search filter. Enter the search filter using standard LDAP protocol.

You can specify one or more attributes to search for. To search all attributes, enter an asterisk (*).

For more information about the LDAP protocol, see your directory documentation.

Search Attributes

Directory Attribute Select the attributes to search for.

Leave this field blank to search all attributes.

Viewing PeopleSoft Directory Interface Reports

This section discusses how to:

- View the directory audit report.
- View the business interlink status report.

Pages Used to View PeopleSoft Directory Interface Reports

Page Name	*Definition Name*	*Navigation*	*Usage*
Directory Audit Report	EO_RUNCTL_DS_AUDIT	Enterprise Components, Directory Interface, Audit Report	Generate the Directory Audit report.
Directory BI Status Report	EO_RUN_DS_BI_RPT	Enterprise Components, Directory Interface, Bus Interlink Status Report	Generate the Directory BI Status report.

Directory Audit Report Page

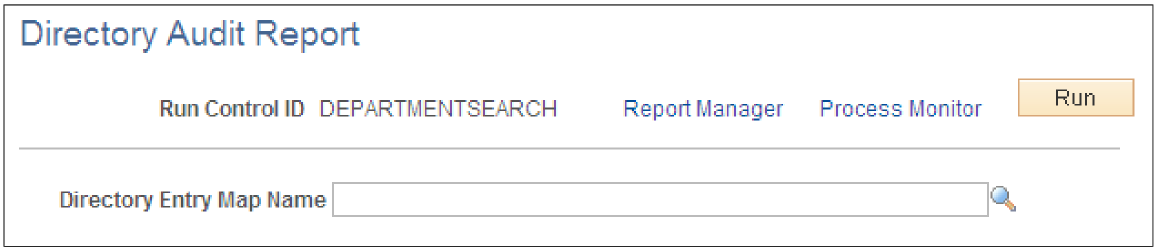
Use the Directory Audit Report page (EO_RUNCTL_DS_AUDIT) to generate the Directory Audit report.

Navigation

Enterprise Components, Directory Interface, Audit Report

Image: Directory Audit Report page

This example illustrates the fields and controls on the Directory Audit Report page. You can find definitions for the fields and controls later on this page.



Directory Audit Report

Run Control ID DEPARTMENTSEARCH

Directory Entry Map Name

Use this page to generate the Directory Audit report.

The Directory Audit SQR report (EO_DS001) locates and reports discrepancies between the PeopleSoft database and your directory. Before you can generate the Directory Audit report, run the Directory Audit process. The Directory Audit process populates a comparison record containing the data that differs between the PeopleSoft database and the directory and creates an LDIF file with this data that can be used to update the directory. The Directory Audit report is based on this record, so you can verify what will be updated in the directory if you apply the LDIF file.

The report generates the following error messages:

- The distinguished name is not found in the directory.
- The distinguished name is not found in PeopleSoft.
- The attribute is in PeopleSoft but not in the directory.
- The attribute is in the directory but not in PeopleSoft.

- The value is in PeopleSoft but not in the directory.
- The value is in the directory but not in PeopleSoft.

Directory BI Status Report Page

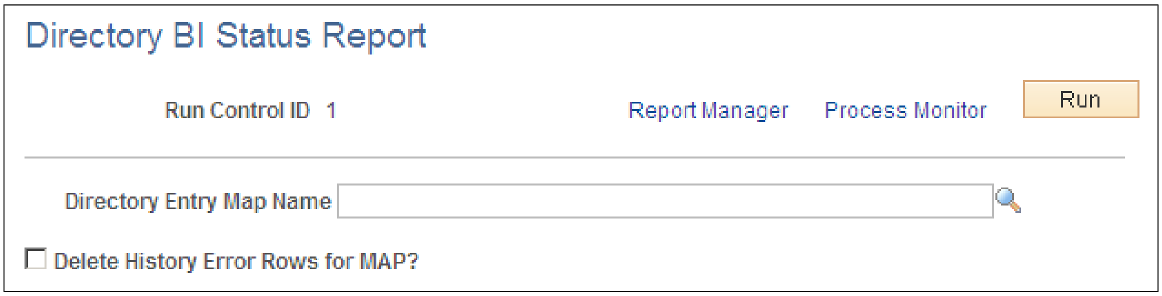
Use the Directory BI Status Report page (EO_RUN_DS_BI_RPT) to generate the Directory BI Status report.

Navigation

Enterprise Components, Directory Interface, Bus Interlink Status Report

Image: Directory BI Status Report page

This example illustrates the fields and controls on the Directory BI Status Report page. You can find definitions for the fields and controls later on this page.



Directory BI Status Report

Run Control ID 1

Directory Entry Map Name

☐ Delete History Error Rows for MAP?

Use this page to generate the BI Status report.

If you selected an output type of Business Interlinks when setting up maps to associate PeopleSoft fields with directory attributes, the system uses PeopleSoft Business Interlinks to modify the directory. If errors occur as a result of the interlinks, the system writes the errors to an error record. The Business Interlink Status SQR report (EO_DS002) retrieves and presents the data contained in this error record.

Delete History Error Rows for MAP Select to delete historical error rows for this map after reporting them. The PS_EO_BILOAD_ERR record retains error data for this map until you run the report with this check box selected.

Run Click Run to generate the report using PeopleSoft Process Scheduler.

For more information, refer the product documentation for *PeopleTools: PeopleSoft Process Scheduler*.

Managing Transaction Audit History

This section discusses how to:

- Generate a transaction history report.
- Search for a transaction history.
- Purge transaction history.

Pages Used to Manage Transaction Audit History

Page Name	***Definition Name***	***Navigation***	***Usage***
Transaction History Report	EODS_RUN_AUDIT	Enterprise Components, Directory Interface, Audit History, Run Report	Generate a transaction history report.
Transaction History Inquiry	EODS_AUDIT_HIS_IN	Enterprise Components, Directory Interface, Audit History, Transaction Inquiry	Search for a transaction history.
Purge Transaction History	EODS_RUN_ADT_PURGE	Enterprise Components, Directory Interface, Audit History, Purge Transactions	Purge audit history data.

Transaction History Report Page

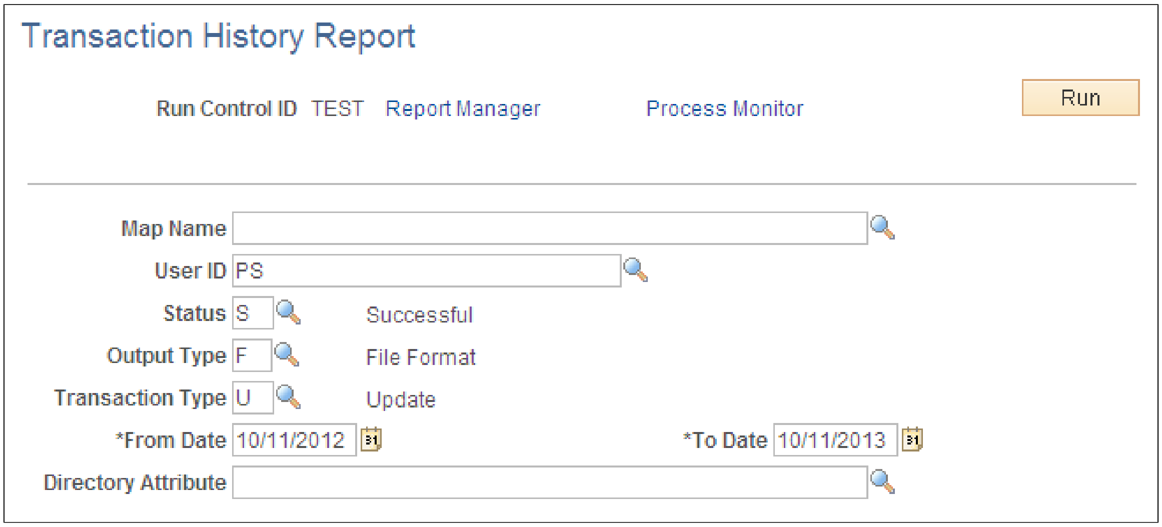
Use the Transaction History Report page (EODS_RUN_AUDIT) to generate a transaction history report.

Navigation

Enterprise Components, Directory Interface, Audit History, Run Report

Image: Transaction History Report page

This example illustrates the fields and controls on the Transaction History Report page. You can find definitions for the fields and controls later on this page.



Transaction History Report

Run Control ID TEST

Map Name

User ID PS

Status S Successful

Output Type F File Format

Transaction Type U Update

*From Date 10/11/2012 *To Date 10/11/2013

Directory Attribute

Use this page to generate a Transaction History report.

Map Name	Select a directory entry map name.
User ID	Select the user ID.
Status	Select the status of the transaction. Valid values are *Successful* and *No Success.*

Output Type

Select the type of output. Valid values are *File Format* and *Business Interlink.*

Transaction Type

Select a type of transaction. Valid values are *Add, Delete,* and *Update.*

Directory Attribute

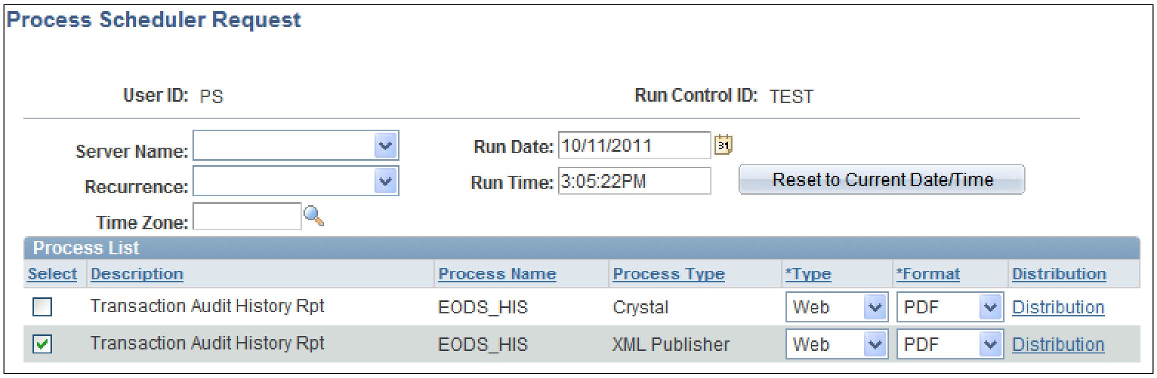
Select the attribute of the entry being defined that will contain all the members of this entry. This attribute must be valid for the current entry in the directory.

The employees who satisfy the entry membership rules of this entry are added under this entry as a new value of this attribute. Because of this, as many attribute values will exist as employees satisfying the entry membership rules. If this field is left blank, the application uses *member* as a default attribute name.

The following example shows the Process Scheduler Request page with Transaction Audit History Report in the process list. The system supports the creation of this report using Crystal Reports or Oracle Business Intelligent Publisher (BI Publisher or BIP).

Image: Process Scheduler Request page showing the Transaction Audit History Report in the process list

This example illustrates the fields and controls on the Process Scheduler Request page showing the Transaction Audit History Report in the process list. You can find definitions for the fields and controls later on this page.



Process Scheduler Request

User ID: PS

Run Control ID: TEST

Server Name:

Run Date: 10/11/2011

Recurrence:

Run Time: 3:05:22PM

Time Zone:

Process List

Select	Description	Process Name	Process Type	*Type	*Format	Distribution
□	Transaction Audit History Rpt	EODS_HIS	Crystal	Web	PDF	Distribution
☑	Transaction Audit History Rpt	EODS_HIS	XML Publisher	Web	PDF	Distribution

Transaction History Inquiry Page

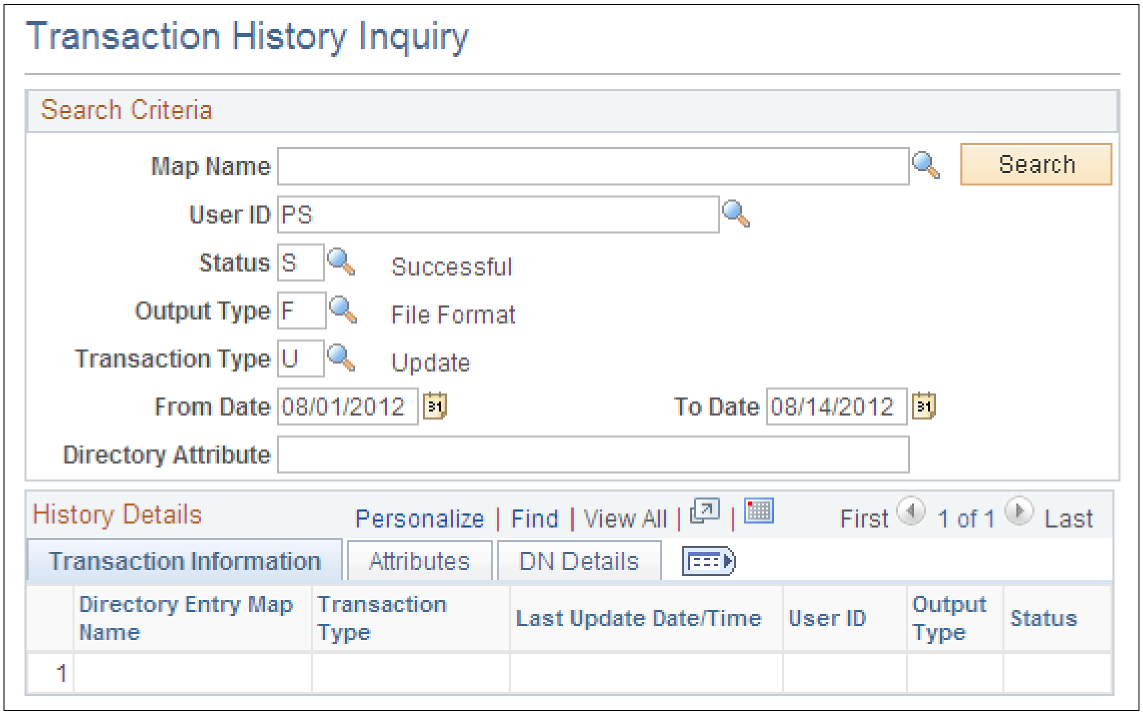
Use the Transaction History Inquiry page (EODS_AUDIT_HIS_IN) to search for a transaction history.

Navigation

Enterprise Components, Directory Interface, Audit History, Transaction Inquiry

Image: Transaction History Inquiry page

This example illustrates the fields and controls on the Transaction History Inquiry page. You can find definitions for the fields and controls later on this page.


Transaction History Inquiry

Search Criteria

Map Name	
User ID	PS
Status	S Successful
Output Type	F File Format
Transaction Type	U Update
From Date	08/01/2012
To Date	08/14/2012
Directory Attribute	

	Directory Entry Map Name	Transaction Type	Last Update Date/Time	User ID	Output Type	Status
1						

Use this page to run a query that appears in the browser rather than creating a report that runs using Process Scheduler.

Map Name

Select a Directory Entry Map name.

User ID

Select the user ID.

Status

Select the status of the transaction. Valid values are *Successful* and *No Success.*

Output Type

Select the type of output. Valid values are *File Format* and *Business Interlink.*

Transaction Type

Select a type of transaction. Valid values are *Add, Delete,* and *Update.*

Directory Attribute

Select the attribute of the entry being defined that will contain all the members of this entry. This attribute must be valid for the current entry in the directory.

The employees who satisfy the entry membership rules of this entry are added under this entry as a new value of this attribute. Because of this, as many attribute values exist as employees satisfying the entry membership rules. If this field is left blank, the application uses *member* as a default attribute name.

Transaction Information Tab

The Transaction Information tab displays information about the transaction including the map name, the status, output type, and transaction type.

Attributes Tab

The Attributes tab displays information about the transaction including what directory audit action was performed, the type of directory attribute, and the attribute value.

DN Details Tab

The DN Details tab displays the distinguished name of the transaction.

Purge Transaction History Page

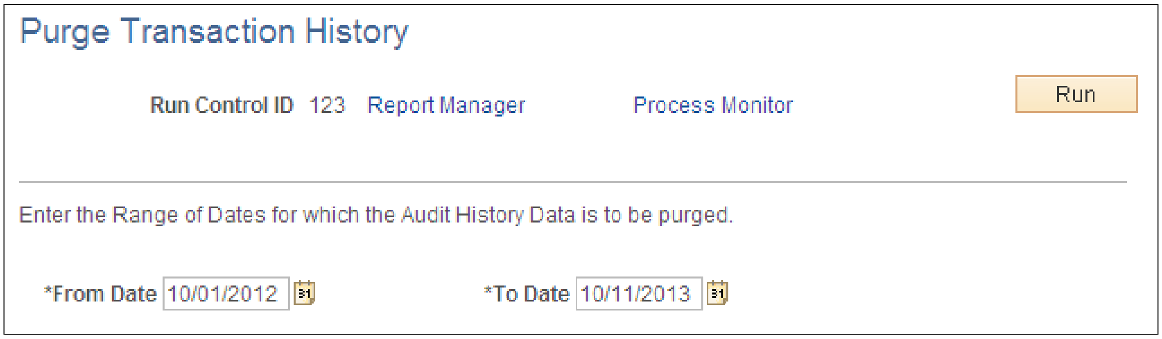
Use the Purge Transaction History page (EODS_RUN_ADT_PURGE) to purge audit history data.

Navigation

Enterprise Components, Directory Interface, Audit History, Purge Transactions

Image: Purge Transaction History page

This example illustrates the fields and controls on the Purge Transaction History page. You can find definitions for the fields and controls later on this page.


Purge Transaction History

Run Control ID 123

Enter the Range of Dates for which the Audit History Data is to be purged.

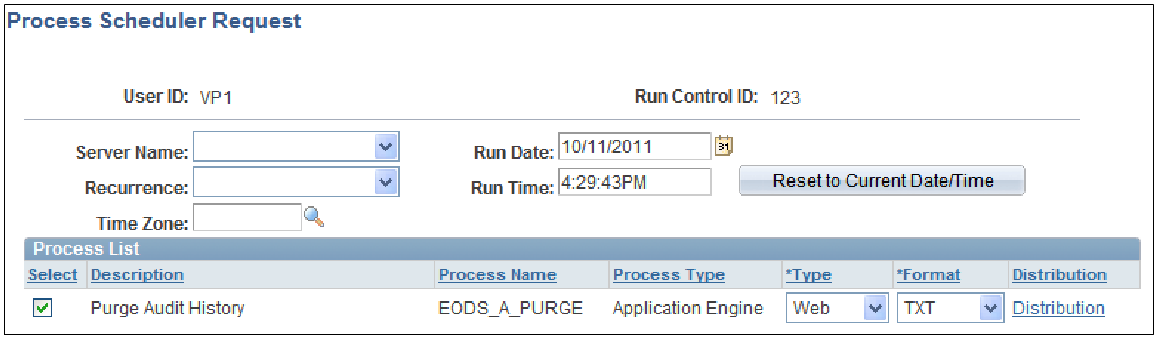
*From Date 10/01/2012

*To Date 10/11/2013

The following shows the Process Scheduler Request page with Purge Audit History in the process list:

Image: Process Scheduler Request page showing Purge Audit History in the process list

This example illustrates the fields and controls on the Process Scheduler Request page showing Purge Audit History in the process list. You can find definitions for the fields and controls later on this page.



Process Scheduler Request

User ID: VP1

Run Control ID: 123

Server Name:

Recurrence:

Time Zone:

Run Date: 10/11/2011

Run Time: 4:29:43PM

Process List

Select	Description	Process Name	Process Type	*Type	*Format	Distribution
☑	Purge Audit History	EODS_A_PURGE	Application Engine	Web	TXT	Distribution

Use the Purge Transaction History page to purge audit history data based on the date range that you enter in the run control page.

Chapter 4

Using the Error Handling Utility

Understanding the Error Handling Utility

The Error Handling utility is a PeopleTools application that you use to view and correct messages that are received by the subscriber. You can also use this utility to correct data that is stored in staging tables.

Warning! Effective with the PeopleTools 8.48 release, the Integration Broker Service Operations Monitor supersedes the Error Handling utility.

For more details, see product documentation for *PeopleTools: Integration Broker Service Operations Monitor*.

Error Management Process

PeopleSoft applications that receive flat file data from other systems through batch processes have built-in facilities to validate and correct data prior to updating the main application tables. Likewise, before updating core PeopleSoft application tables, the subscription process detects data errors in the messages that it receives. These error messages are stored in either message queues or staging tables.

In some cases, however, errors can't be sent back to the third party for correction (such as when data is in a flat file). In these cases, you must provide error processing on incoming data so that messages that contain information about business objects, such as items and vendors, can be corrected and reprocessed in the PeopleSoft system.

In many integration point implementations, especially those that involve huge transactions and complex data validations, subscription codes are written to run simple incoming data validations. Upon a successful outcome, the system performs these steps:

1. Writes incoming data to staging tables.

2. Runs more stringent data validation processes, usually in batch processes.

As a result of the first subscription process, entries are written to a special staging table (EO_EIP_CTL). The EO_EIP_CTL table is keyed by a single key, EIP_CTL_ID, and has messaging keys. The EO_EIP_CTL table maintains links between the source message and the staged data and ensures that the data maintenance program can identify the staging tables.

To find the particular message that you want to view:

1. Select a message by using standard selection criteria, including data maintenance programs, business unit or setID, data status, and staging table or message queue.

2. Click a detail button to access the application page related to the edited message.

3. Make your edits.

4. Save the message.

Note: Messages that are selected from a staging table are saved to the staging table. Before they're saved permanently (to the application table), the corrected messages that were selected from the message queue are assigned a status of *Changed* and saved to the same queue. The messages then undergo the subscription process data validation again, after which they're saved to an application table.

A data maintenance program corrects errors in data that enters the PeopleSoft system.

Note: Subscription error management does not apply to real-time data that enters the PeopleSoft system through PeopleSoft Business Interlinks.

Error management consists of the following sequential activities:

1. The PeopleSoft system receives XML messages from a third-party source.
2. Subscription processes validate the data and send the data to:
   - PeopleSoft application tables (if no errors were found).
   - Message queue (if errors were found).
   - PeopleSoft staging tables (if subscription logic requires).
3. The Error Handling utility interacts with:
   - Message queue.
   - Staging tables.
4. The user interacts with:
   - The Error Handling page (to select a message for review or correction).
   - Application pages (for editing data errors).
5. The PeopleSoft system updates:
   - Message queue (if the original message was selected from the queue).
   - Staging tables (if the original data was extracted for staging tables).

Setting Up and Maintaining Message Errors

This section discusses how to:

- Create error-correction pages.
- Set up the error handling utility.

- Set up row security.
- Set up workflow notification in PeopleSoft Application Designer.
- Test the error handling utility.

Warning! Effective with the PeopleTools 8.48 release, the Integration Broker Service Operations Monitor supersedes the Error Handling utility.

Creating Error-Correction Pages

Use PeopleSoft Application Designer to create error-correction pages and components.

Note: The Header Details page and the Error Details page are applicable only if data resides in the messaging queues. Use one Header Details page and one Error Details page for each field.

When creating error-correction pages, follow these guidelines:

- Use EO_EIP_CTL as the main header table if you use staging tables.

  Developers can use unique EIP_CTL_ID generator functions (Generate_EIP_CTL_ID and Increment_EIP_CTL_ID) available in the FUNCLIB_EOEIP.EIP_CTL_ID FieldFormula event.

- Use the generate EIP_CTL_ID function to generate a unique key value for EIP_CTL_ID, based on the subscription process instance.

  The EIP_CTL_ID function is the sole key in the EO_EIP_CTL header table.

- Hook to EO_EIP_CTL to use the Error Handling utility for staging-table-based error handling.

  See Data Maintenance Page.

- Modify the record PeopleCode in the derived work record EO_EIP_CTL_WRK.

  The derived work record called EO_EIP_CTL_WRK contains most of the processing logic for the utility. Although most of the codes are generic, you must write application-specific codes for the return button field (EIP_RETURN_BTN).

- Add codes to return the user to the main Error Handling page.

  Use the EO_EIP_CTL_WRK derived record in your component and write component PeopleCode for the record to handle any unique requirements.

  Note: This unique code belongs only to your component and cannot be shared.

FUNCLIB_XXEIP Codes

Functions are created and stored in product-specific FUNCLIB_XXEIP records, with *XX* representing the product. Application developers can look up these function libraries for possible use as templates.

Common integration-related functions are placed in the FUNCLIB_EOEIP record.

EIP_DETAIL_BTN FieldFormula

Function Copy_Detail_Errors (&WRK_FIELD, &J, &WRK_ROWSET) copies edit errors to a generic error subpage.

Function BuildQueueRowset (&WRK_ROWSET1 As Rowset, &WRK_ROWSET2 As Rowset, &MSG_ROWSET As Rowset, &SCROLL, &RECORD_FROM, &RECORD_TO) generically builds the queue-based transfer page. This function is useful for single record messages.

FUNCLIB_EOEIP.EIP_CTL_ID FieldFormula

You need an EIP_CTL_ID every time that you process a message to maintain error handling. Use Function Generate_EIP_CTL_ID and Increment_EIP_CTL_ID to create EIP_CTL_IDs based on a new subscription process instance. For the Generate_EIP_CTL_ID function, invoke method 4 from the list of process instances.

Note: Although in some cases a random number generator is used for creating a new EIP_CTL_ID by calling Generate_EIP_CTL_ID(1), you should use the subscription process instance if possible. Generate_EIP_CTL_ID(4) uses the identifier number of the message subscription process. Remember to activate this on the message subscription properties by selecting the Generate Subscription Process Instance check box.

Warning! The Generate_EIP_CTL_ID and Increment_EIP_CTL_ID functions call other functions in FUNCLIB_ININTFC (an FDM record). You must re-create these additional functions whenever you use Generate_EIP_CTL_ID and Increment_EIP_CTL_ID within applications.

Data Maintenance Page

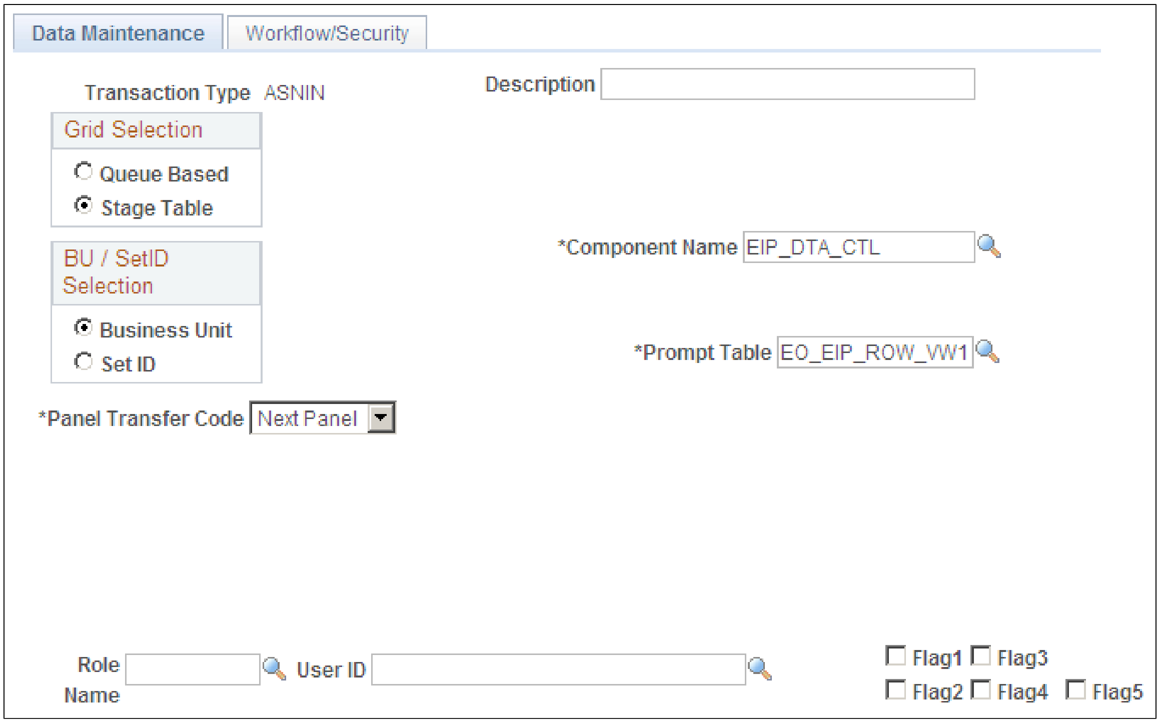
Use the Data Maintenance page (EO_EIP_CTL_SETUP1) to set up the Error Handling utility and correct message errors.

Navigation

Enterprise Components, Integration Definitions, Data Maintenance Utility

Image: Data Maintenance page

This example illustrates the fields and controls on the Data Maintenance page. You can find definitions for the fields and controls later on this page.



Transaction Type ASNIN

Description

Grid Selection
○ Queue Based
● Stage Table

BU / SetID Selection
● Business Unit
○ Set ID

*Component Name EIP_DTA_CTL

*Prompt Table EO_EIP_ROW_VW1

*Panel Transfer Code Next Panel

Role Name

User ID

□ Flag1 □ Flag3
□ Flag2 □ Flag4 □ Flag5

Use this page to set up the Error Handling utility and correct message errors.

Queue Based and Stage Table

PeopleTools messages are queue-based; bulk data or application data is usually staged.

Business Unit and SetID

Select a business unit or setID on which to base the data.

Component Name

Select the name of the component for accessing the Error Handling/Data Maintenance page (EO_EIP_CTL_MAINT). This is the component that the user opens for error-correction activities. You should select *EIP_DTA_CTL.* This component uses the same fonts and images for all incoming data maintenance or error-correction activities within the PeopleSoft system for non-real-time integration.

Prompt Table

Follow the navigation path to use the Error Handling utility, and select a prompt table for the data maintenance program (EIP_PROGRAM). The PeopleSoft system includes a common error handling menu structure that points to the main component EIP_DTA_CTL. Based on user privileges, the Error Handling utility prompts from a selected list of data maintenance programs.

Panel Transfer Code

Select *Next Panel* to receive a transfer panel name prompt, which gives you a selection of pages that are part of the main data maintenance component that you selected in the Main Data Maint Panel Group field. Select one page.

Selecting either *Transfer* or *Modal* opens additional fields.

Role Name

Select a name to set up workflow notification and to notify all role users listed.

User ID

Select to notify only specific role users.

Flag1, Flag2, Flag3, Flag4 and Flag5

Users often do not see these check boxes. Application developers use them for extra coding flexibility. Developers must create documentation for these check boxes to use them.

Workflow/Security Page

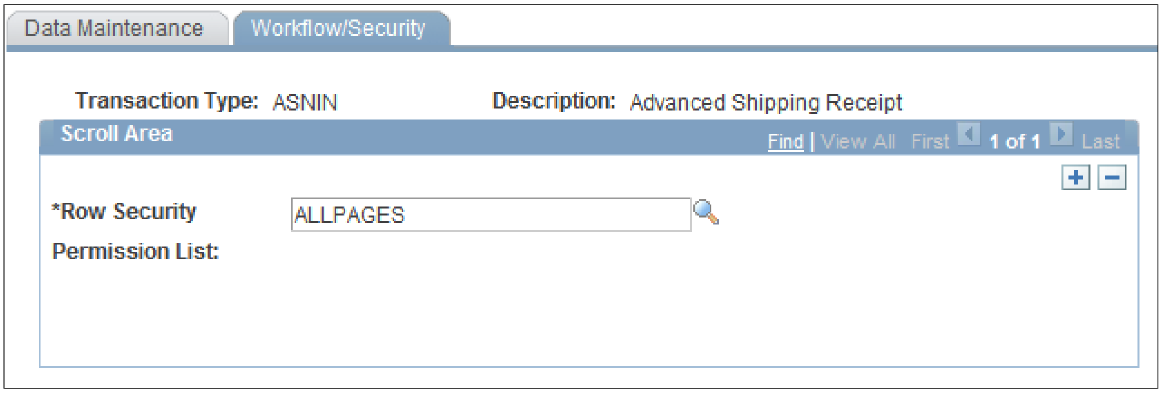
Use the Workflow/Security page (EO_EIP_CTL_SETUP2) to set up row security.

Navigation

Enterprise Components, Integration Definitions, Data Maintenance Utility, Workflow/Security

Image: Workflow/Security page

This example illustrates the fields and controls on the Workflow/Security page. You can find definitions for the fields and controls later on this page.



Transaction Type: ASNIN

Description: Advanced Shipping Receipt

*Row Security Permission List: ALLPAGES

Use this page to set up row security.

Row Security Permission List

Assign a row security class to the transaction type that you are setting up.

Note: The value that you select determines which user can see the given transaction type. You can enter multiple row security classes.

Setting Up Workflow Notification in PeopleSoft Application Designer

You can set up an EO_WF_ERR Application Engine process to scan the EO_EIP_CTL table periodically to look for rows with *ERROR* status. When the system finds errors, it generates workflow notifications and routes them to the role users that you designated in the User ID field on the Data Maint Setup1 page. Clicking the worklist item in the workflow notification transfers the user to the Error Handling/Data Maint (error handling/data maintenance) page.

To set up workflow notification:

1. Open an instance of PeopleSoft Application Designer.
2. Select File, Open.
3. Select *Business Process* from the Definition drop-down menu.
4. Open business process EC_MANAGE_ERRORS.
5. Right-click the MANAGE ERRORS icon and select *View Definition.*
6. Double-click the Correct Errors icon to open the Step Definition dialog box.
7. Click the Attributes button to open the Step Attributes dialog box.
8. Complete the required navigation information to the error correction page in the Step Attributes dialog box and click OK.

Data Maintenance Page

Use the Data Maintenance page (EO_EIP_CTL_MAINT) to test the error handling utility.

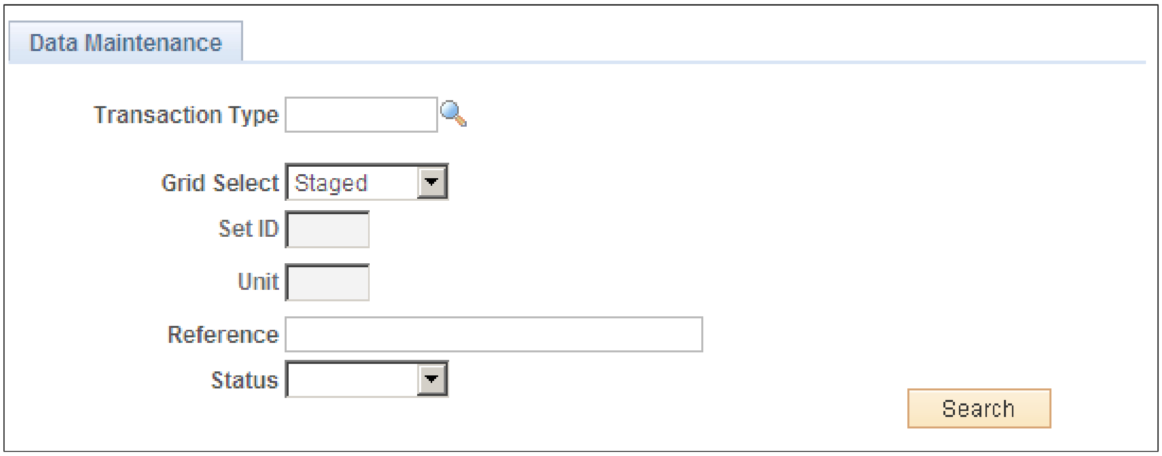
Test subscription processes must error out. Create a message subcontract with Error status, or create a stage-based incoming transaction with a status value of 1 (Error).

Navigation

Enterprise Components, Integration Definitions, Review Centralized Error

Image: Data Maintenance page

This example illustrates the fields and controls on the Data Maintenance page. You can find definitions for the fields and controls later on this page.



Data Maintenance

Transaction Type

Grid Select: Staged

Set ID

Unit

Reference

Status

Use this page to test the error handling utility.

Test subscription processes must error out. Create a message subcontract with *Error* status, or create a stage-based incoming transaction with a status value of 1 *(Error).*

Note: Error conditions depend on processes that are specific to the application.

The Review Centralized Error page features the following page elements:

Transaction Type These values are provided by your application.

Grid Select Select whether the errors you want to check are queue-based, stored in a staging table, or select and search for a specific error.

SetID Enter a setID.

Unit Enter the unit of measure.

Reference Enter a reference name.

Status Select from *Cancelled, Complete, Error, Hold, In Process, Incomplete, New* and *Reprocess.*

▤ Click the Show Detail Entry icon to transfer to the application page for necessary error-correction activities.

Correcting Message Errors

This section provides an overview of the Workflow Notification process and discusses how to correct message errors.

Warning! Effective with the PeopleTools 8.48 release, the Integration Broker Service Operations Monitor supersedes the Error Handling utility.

Understanding the Workflow Notification Process

Workflow error notification involves the following steps:

1. A process-scheduled Application Engine program scans data errors in the staging tables (or in the Subscription Contract Message queues) at a given interval.

   Note: You can also run a separate Workflow Notification Application Engine program for messaging queues.

2. When the Application Engine program finds errors in the actual message queue or the staging summary table, it opens respective component interfaces that invoke the designated temporary components for inserting error data into the underlying temp tables.
3. Saving the temporary page (through component interface calls) triggers workflow PeopleCode to send worklist notifications to designated users, depending on the role name or user ID that you selected on the Data Maint Setup1 page.

Note: The Application Engine program traps only the first error for each field, but it can trap multiple errors for each record.

Correcting Message Errors

To correct message errors:

1. Access the Data Maint Setup1 page to open the worklist.
2. Open the worklist.

   You are transferred to the Data Maintenance page (EO_EIP_CTL_MAINT) to correct the errors.
3. Find the error rows by using a combination of search criteria, such as transaction type, business unit or setID, and error status.
4. Click the Search icon.
5. Click the Show Detail Entry icon (to the left of each row) to select a specific transaction that represents a single message or a single row of header-level record from the staging table.
6. Correct the data.
7. Click the Return button to return to the Error Handling Summary page (EO_EIP_CTL_MAINT) and continue the session with other transactions.

Note: In this scenario, the corrected data does not update core PeopleSoft tables. Instead, it updates the message queue or staging tables.

Chapter 5

Using the Publish Utility

Understanding the Publish Utility

The Publish utility automates the process of copying the contents of an entire table into a remote database or legacy system.

Use the utility to synchronize data from an existing system when a new PeopleSoft system is installed.

The Publish utility is built upon the PeopleSoft Integration Broker services-oriented architecture and publishes service operations to target systems. The service operations published contain messages that store the table data.

Use the Publish utility to perform full table publish or batch publish processing. These process are defined as:

Full table publish The full publish process seeds, or initially populates or repopulates, a copy of an entire table into a remote database or legacy system. The entire contents of the table are published to all systems that require a copy of the table.

Batch publish This term describes jobs or processes that run independently of their initiating process. A batch process can also run at one or more predetermined times in the future from the initiating request. A batch process is appropriate for publishing incremental changes to data in a batch environment or for processing large volumes.

To use the utility you use the PeopleSoft Pure Internet Architecture to create publishing rules and then assign them to service operation messages to control how the utility manages the messages it is publishing.

Understanding Publishing Rules

You control the size and number of, and the frequency at which data full table or batch data is published by creating and using a series of data publishing rules.

The PeopleSoft Pure Internet Architecture component that you use to create publishing rules depends on the type of processing that you want to perform. For full table publish processing, use the Full Table Publish Rules component (EOIU_SOPUBFULL) to create publishing rules; for batch publish processes, use the Batch Process Rules component (EO_MSGPUBBATCH).

The pages in the Full Table Publish Rules and Batch Process Rules components are very similar. You use these components to specify:

- Pre-processing and post-processing tasks.
- The source of the data.
- Whether to chunk the message.
- Record mapping.
- Output format.
- Related languages in which to publish the data. (Full table publishing rules only)

Pre-Processing and Post-Processing

Often, performing pre-processing and post-processing tasks on the subscribing system before or after it processes a service operation message is desirable. Pre-processing tasks can include actions such as deleting existing data. Post-processing actions can include data clean-up.

Performing pre-processing and post-processing tasks is optional. However, if you choose to perform them, you must enable the Message Header or Message Trailer options on the Full Table Publish Rules or Batch Publish Rules page, as well as code the processing logic on the service operation handler.

You can perform pre-processing and post-processing tasks on service operations that are processed sequentially or in parallel.

Note: Using pre-processing and post-processing tasks in conjunction with the parallel processing feature of the Publish Utility is currently available in the HCM application only. To use pre- and post-processing tasks in conjunction with parallel processing you must apply one of the following resolutions to both the publishing and subscribing systems: HCM 9.0 systems – Resolution 827074; HCM 9.1 systems – Resolution 827350.

Using pre-processing and post-processing logic is discussed in greater detail elsewhere in this documentation.

See Performing Pre-Processing and Post-Processing Tasks.

Data Sources

The data to publish using the Publish utility can be from the following sources:

- A staging table.
- A temporary table.
- A view.

Message Chunking

This section discusses message chunking as it pertains to specifying chunking options in publishing rules. Setting up message chunking is discussed elsewhere in this documentation.

See Setting Up Message Chunking.

Chunking

Chunking refers to the automatic breaking of a message into several smaller messages based on values in fields in the level zero record.

You can perform chunking based on setID, business unit, or record group.

Chunking on business unit means that all transactions within the message are for the same business unit value.

The system chunks message data based on the MaxMessageSize parameter.

The MaxMessageSize (maximum message size) field in the PeopleSoft Option (PSOPTIONS) table limits the size of the message. Before processing each level zero record, the Full Table Publish utility compares the size of the message against the value in the MaxMessageSize field. When the message size exceeds the value in the MaxMessageSize field, the service operation publishes, and a new service operation starts.

You can also specify message chunking in the publish rules for the service operation, which enables the message to publish when the value of a chunk field changes.

Chunking Field

Chunking fields are key fields in the level zero record that are used to break the message into parts.

Chunking Table

A chunking table is a derived or Structured Query Language (SQL) table that contains the fields by which the message is chunked. SQL chunk tables define the valid values of the chunking fields and the nodes to which the message is published.

Oracle provides three standard chunking tables: EO_BUSUNIT_EOC for business unit values, EO_SETID_EOC for setID values and EO_RECGRP_EOC for record groups values. Oracle provides chunking tables for business unit, setID and record group that are maintained by a series of components (such as components that are created for maintaining the business unit chunking table).

Alternate Chunking Table

An *alternate chunking table* is a secondary chunk table. It provides a separate view of an existing chunk table, but with one or more field names customized. It enables you to reuse an existing chunk table. For example, if the record EO_BUSUNT_EOC has BUSINESS_UNIT as the chunking field, you can create a view of this table that has BUSINESS_UNIT_IN as the chunking field.

When the field being chunked doesn't use the normal field names for business unit and setID, the alternate chunk table enables you to use the existing chunk tables and maintenance pages for business unit and setID. The alternate chunk table enables the user to use the existing chunk tables and maintenance pages for business unit and setID when the field being chunked doesn't use the normal field names for business unit and setID, for example, BUSINESS_UNIT_IN. Use the _EOV suffix for alternate chunk tables.

Chunking Rule

A chunking rule points to the chunking table. Multiple chunking rules can point to the same chunking table.

Record Mapping

You should use the record mapping feature in a publish rule because:

- The published data comes from staging tables or temporary tables.

  Using staging tables is the only way to publish rows that have been deleted from application master tables.

- The published data needs an order sequence that differs from that of the records in the message.
- You must create a view that selects only current effective-dated rows in situations in which the application table is effective-dated and the subscribing database cannot process future-dated transactions.

Output Format

You can choose the output of the Full Table Publish utility to be in the following formats:

- Message (default value).
- Flat file.
- Flat file with control record.

You can create multiple publish rules with different output formats for the same service operation.

Flat File Output

You select the flat file option when you set up publish rule definitions for the service operation. If a file layout object exists with the same name as the message object, you can modify the output format field on the Publish Rule Definition page.

If you select flat file output, the header and trailer messages aren't created, and a single output flat file is created even if you already specified the chunking rule.

The file layout definition must have a record structure identical to that of the message; if the AUDIT_ACTN field does not exist in the record, you must add it to the file layout record definition.

The option becomes modifiable if a file layout object exists with the same name as the message definition.

The directory location of the flat file is determined by the value in the OUTPUT parameter in PeopleSoft Configuration Manager.

The flat file name is *messageName_SequenceNumber*.out, where *messageName* is the message name and *SequenceNumber* is the next available sequential number for files with that message name.

Publishing Data in Related Languages

The Publish utility provides the option to publish data in related languages.

If you choose to use the related language feature, the system creates the following related language messages:

- One message in the base language of the publishing system.
- One message for each language in the related language tables for the base tables.

The subscribing system receives the messages in the order in which they are published. For example, if the base language is English, with French and then German as related language tables, the Publish utility creates the messages in this order:

1. (Optional) Header message.
2. English message 1.
3. French message 1.
4. German message 1.
5. English message 2.
6. French message 2.
7. German message 2.
8. (Optional) Trailer message.

Prerequisites for Using the Publish Utility

The Publish Utility uses the PeopleSoft Integration Broker services-oriented architecture and to publish service operations to target systems. As such, to use the utility you must have:

- An understanding of PeopleSoft Integration Broker.
- PeopleSoft Integration Broker configured for your integration requirements.
- Service operations, including messages, handler, and routing definitions, setup on sending and receiving systems.
- Using pre-processing and post-processing tasks in conjunction with the parallel processing feature of the Publish Utility is currently available in the HCM application only. You must apply one of the following resolutions to both publishing and subscribing systems:
    - HCM 9.0 systems: Resolution 827074.
    - HCM 9.1 systems: Resolution 827350.

Common Elements Used in The Publish Utility

Alternate Chunk Table

(Optional.) Enter the name of an alternate chunk table.

This field in disabled until you enter a value in the Chunk Rule ID field.

Service Operation

Click the link to view the service operation definition associated with the publish rule.

Chunking Rule ID

Associate a chunking method with the publish rule, if needed.

Message Options

The valid options are:

- Create Message Header

  Select the check box to have the system create a message header when the publish rule is invoked. Clear the check box if you do not want a message header created when the publish rule is invoked.

  By default, the Create Message Header check box is selected.

- Create Message Trailer

  Select the check box to have the system create a message trailer when the publish rule is invoked. Deselected the check box if you do not want a message trailer created when the publish rule is invoke.

  By default, the Create Message Trailer check box is selected.

Publish Rule ID

Enter the name of the publish rule to create and assign to the message.

(Publish Rule ID) Description

Enter a description for the publish rule.

Message.Version

Displays the message and message version associated with the service operation assigned to the publish rule.

Output Format

Indicates the output format of the Publish utility.

The valid output formats are:

- Message (default value).
- Flat file.
- Flat file with control record.

See Understanding Publishing Rules.

Record Source Mapping

Use this section to specify the source data for a record in a message.

The page elements in this section are:

- Message Record Name: Enter the name of the record in the message that you want to map to another record.

- Source/Order By Record Name: Enter the record name that the Publish utility uses to select data.

Service Operation

Displays the name of the service operation associated with the publish rule.

(Service Operation) Description

Displays the description of the service operation associated with the publish rule.

Status

The valid values are:

- *Active:* Activates the publish definition.
- *Inactive:* (Default) Inactivates the publish definition.

Using Sequential and Parallel Processing

This section discusses how to:

- Use sequential processing.
- Use parallel processing.
- Changing processing modes.

Understanding Using Sequential and Parallel Processing

Subscribing systems can process service operations sequentially or in parallel.

In *sequential processing*, the subscribing system processes messages defined in inbound service operations in the order received.

In *parallel processing*, the subscribing system processes messages defined in service operations in parallel, in no specific sequence.

You can use sequential or parallel processing when using either full table publishing or batch publishing.

You configure sequential or parallel processing at the queue level on the subscribing system.

Prerequisites for Using Parallel Processing

Note the following prerequisites for using pre-processing and post-processing tasks in conjunction with the parallel processing feature of the utility:

- The pre-processing and post-processing task functionality of the parallel processing feature of the Publish Utility is currently available in the HCM application only.
- To use pre-processing and post-processing tasks in conjunction with parallel processing you must apply the following resolutions to both the publishing and subscribing systems:
  - HCM 9.0 systems: Resolution 827074.

- HCM 9.1 systems: Resolution 827350.

Using Sequential Processing

To implement sequential processing, on the subscribing system associate the message on the inbound service operation with an *ordered* queue.

See *PeopleTools: PeopleSoft Integration Broker*,“Managing Service Operations,” Defining Service Operation Version Information, “Specifying Messages for Service Operations”

Using Parallel Processing

To implement parallel processing, on the subscribing system associate the message on the inbound service operation with an *unordered* queue. If the message is currently assigned to an ordered queue, assign it to a unordered queue.

In order to realize any performance gain from parallel processing, multiple subscription handlers must be set up for the subscription node to process messages on the unordered queue.

Only Messages that utilize functions inside the PeopleCode function library FUNCLIB_EOEIP for their pre-processing, post-processing and subscription handler logic will have the order of processing logic enforced when using the parallel processing mode.

See *PeopleTools: PeopleSoft Integration Broker*, “Managing Service Operations,” Defining Service Operation Version Information, “Specifying Messages for Service Operations”

See *PeopleTools: PeopleSoft Integration Broker*, “Managing Service Operation Queues,” Adding Queue Definitions

Changing Processing Modes

If your business requirements change, you can change the processing mode from sequential processing to parallel processing or vice versa, by moving the message associated with the service operation to the appropriate queue type.

If you have coded pre-processing or post-processing tasks, you must move the code for those tasks in the service operation message. More information is provided elsewhere in this documentation.

See Performing Pre-Processing and Post-Processing Tasks.

Assigning Full Table Publishing Rules

All PeopleSoft applications use common, centralized tables and pages to define how to publish full table messages. The Publish utility uses publishing rules to process the data.

This section discusses how to:

- Associate full table publishing rules with messages.
- Map a full table message record to another record.

- Specify languages in which to publish messages.

Note: You can create multiple publish rules for the same message. The Full Table Publish utility treats each publish rule as a separate publishing cycle.

Pages Used to Assign Full Table Publishing Rules

Page Name	*Definition Name*	*Navigation*	*Usage*
Full Table Publish Rules	EOIU_SOPUBFULL	Enterprise Components, Integration Definitions, Full Data Publish Rules	Associate a rule with a message and characterize the rule.
Record Mapping	EOIU_SORECMAP	Enterprise Components, Integration Definitions, Full Data Publish Rules, Record Mapping	Map a message record to another record.
Languages	EOIU_SOLANGUAGE	Enterprise Components, Integration Definitions, Full Data Publish Rules, Languages	Specify languages in which to publish a message.

Full Table Publish Rules Page

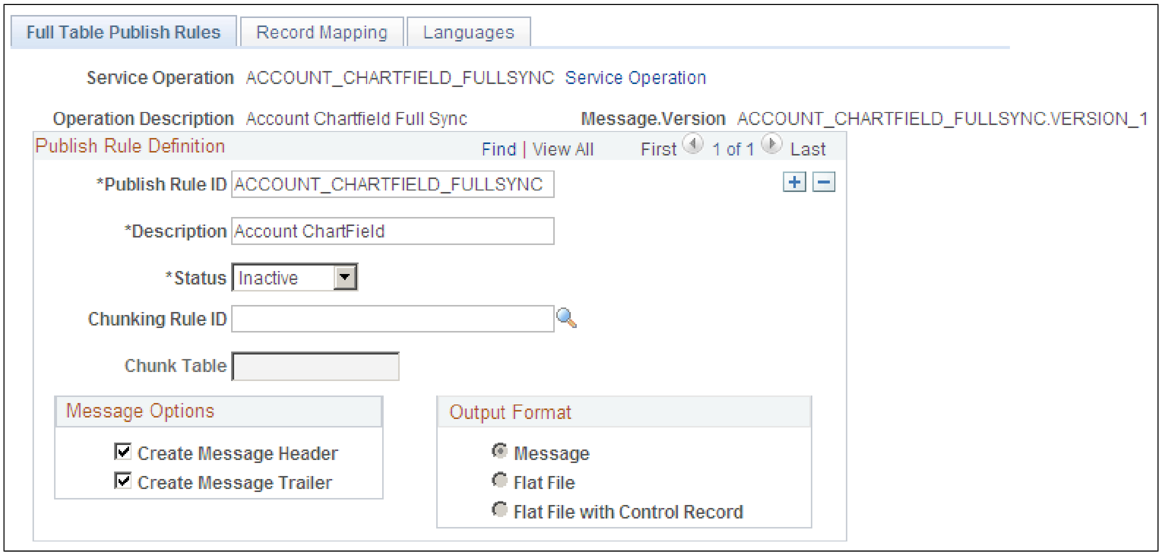
Use the Full Table Publish Rules page (EOIU_SOPUBFULL) to associate a rule with a message and characterize the rule.

Navigation

Enterprise Components, Integration Definitions, Full Data Publish Rules

Image: Full Table Publish Rules page

This example illustrates the fields and controls on the Full Table Publish Rules page. You can find definitions for the fields and controls later on this page.



Service Operation ACCOUNT_CHARTFIELD_FULLSYNC Service Operation

Operation Description Account Chartfield Full Sync

Message.Version ACCOUNT_CHARTFIELD_FULLSYNC.VERSION_1

*Publish Rule ID ACCOUNT_CHARTFIELD_FULLSYNC

*Description Account ChartField

*Status Inactive

Chunking Rule ID

Chunk Table

☑ Create Message Header
☑ Create Message Trailer

● Message
○ Flat File
○ Flat File with Control Record

Use this page to associate a rule to a message and characterize the rule.

Message Options

The valid values are:

- *Create Message Header*. Use to initiate pre-processing tasks on the subscribing system.
- *Create Message Trailer*. Use to initiate post-processing tasks on the subscribing system.

By default, both options are enabled.

Ensure that the subscribing process does not need the header or trailer process before you deselect these check boxes for a Batch Publish message.

When sending messages sequentially, header messages trigger special logic (in a PeopleCode program) on the PeopleSoft full message subscription that deletes the existing application records. Also, some applications use the trailer message to indicate that all data messages have been received and to initiate the validation process. The documentation for the individual message should note whether headers and trailers are supported.

Create Delay Records

This check box appears only if the message name ends with *FULLSYNC_EFF* (such as MESSAGENAME_FULLSYNC_EFF). Select this check box to write all future-dated rows to the delay table. Also select this check box in conjunction with record mapping views that publish only the current effective-dated rows.

Record Mapping Page

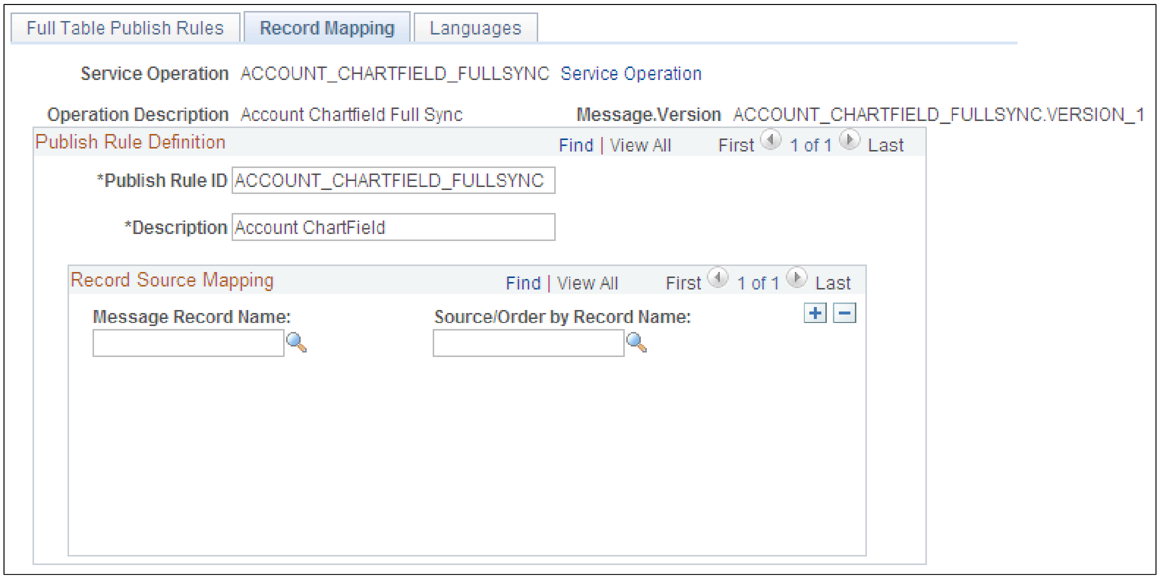
Use the Record Mapping page (EOIU_SORECMAP) to map a message record to another record.

Navigation

Enterprise Components, Integration Definitions, Full Data Publish Rules, Record Mapping

Image: Record Mapping page

This example illustrates the fields and controls on the Record Mapping page. You can find definitions for the fields and controls later on this page.



Service Operation ACCOUNT_CHARTFIELD_FULLSYNC Service Operation

Operation Description Account Chartfield Full Sync Message.Version ACCOUNT_CHARTFIELD_FULLSYNC.VERSION_1

*Publish Rule ID ACCOUNT_CHARTFIELD_FULLSYNC

*Description Account ChartField

Message Record Name:

Source/Order by Record Name:

Use this page to map a message record to another record.

Regardless of which source table you use (staging table, temporary table, or a view), ensure that the source table field names are identical to the field names in the target message record. Key fields must also adhere to the parent and child relationship. (Keys of a parent record must exist in the child record, in the same sequence.) The Publish utility uses the Source/Order By Record Name field to select rows for publishing. You use key matching to find all child rows of a parent.

The chunk field should be the primary key field in all Source/Order By records. If the chunk field is not a key field of the level zero record, join the chunk field to all records in which the chunk field does not exist by using views in PeopleSoft Application Designer.

Enter only those message records with a different source or ordering record. If the message record name and the source or ordering record name are identical, do not insert a row for that record on the Record Mapping page.

Languages Page

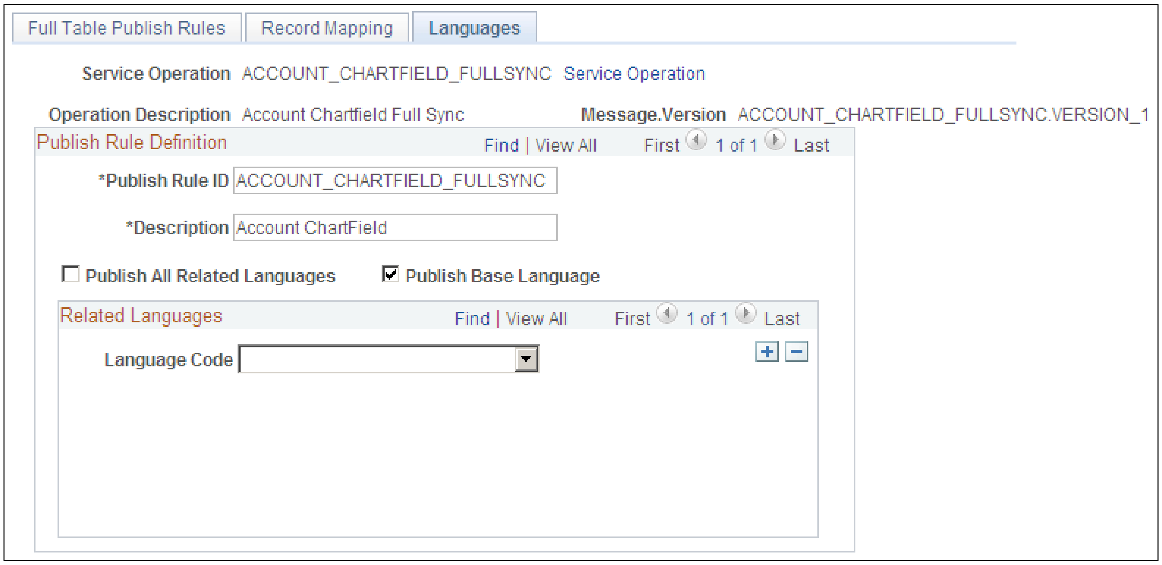
Use the Languages page (EOIU_SOLANGUAGE) to specify languages in which to publish a message.

Navigation

Enterprise Components, Integration Definitions, Full Data Publish Rules, Languages

Image: Languages page

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



Service Operation ACCOUNT_CHARTFIELD_FULLSYNC

Operation Description Account Chartfield Full Sync

Message.Version ACCOUNT_CHARTFIELD_FULLSYNC.VERSION_1

*Publish Rule ID ACCOUNT_CHARTFIELD_FULLSYNC

*Description Account ChartField

□ Publish All Related Languages ☑ Publish Base Language

Language Code

Use this page to specify languages in which to publish a message.

Publish All Related Languages	Select to indicate whether to publish the message in all of the related languages. If this option is selected, the scroll area that you use to enter individual related languages is unavailable. This check box is deselected by default.
Publish Base Language	Select to publish the message in the base language. This check box is selected by default.
Language Code	Select the related language in which to publish the message.

Assigning Batch Publishing Rules

All applications can use common, centralized tables and pages to define how to publish incremental messages from an application program. The Publish utility uses batch publish rules to process the data from the application program.

This section discusses how to:

- Associate a rule with a message and characterize the rule.
- Map a batch publishing message record to another record.
- Assign an application program to a publishing rule.

Note: You can link application programs to multiple publishing rules for the same message or different messages. The Publish utility treats each publishing rule as a separate publishing cycle.

Batch Publish Rules Page

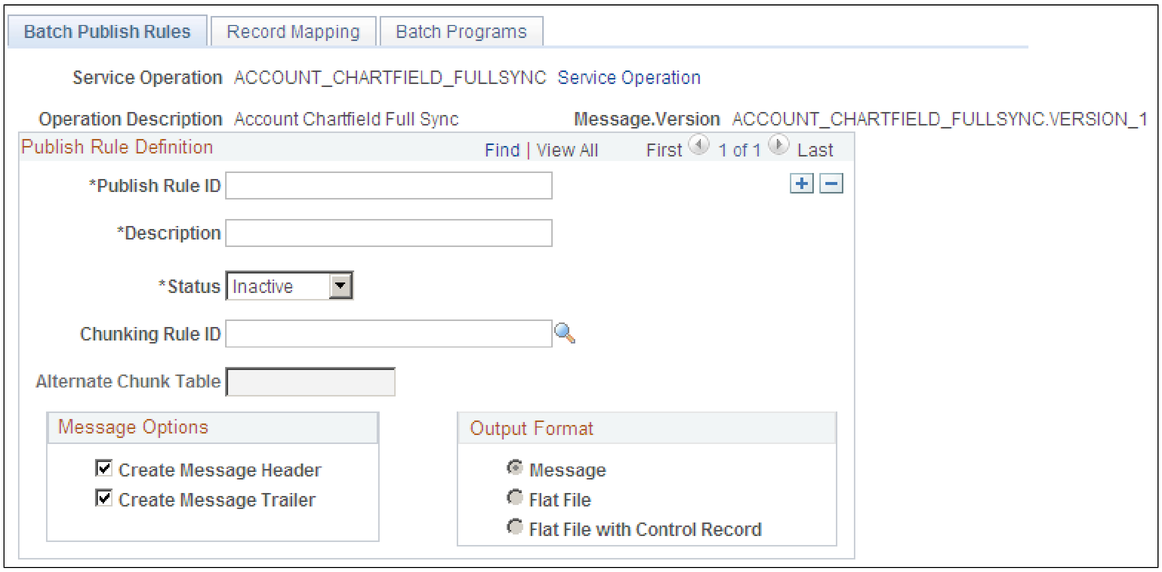
Use the Batch Publish Rules page (EOIU_SOPUBATCH) to associate a rule with a message and characterize the rule.

Navigation

Enterprise Components, Integration Definitions, Batch Publish Rules

Image: Batch Publish Rules page

This example illustrates the fields and controls on the Batch Publish Rules page. You can find definitions for the fields and controls later on this page.



Service Operation ACCOUNT_CHARTFIELD_FULLSYNC Service Operation

Operation Description Account Chartfield Full Sync

Message.Version ACCOUNT_CHARTFIELD_FULLSYNC.VERSION_1

Publish Rule Definition

*Publish Rule ID

*Description

*Status Inactive

Chunking Rule ID

Alternate Chunk Table

Message Options

☑ Create Message Header

☑ Create Message Trailer

Output Format

● Message

○ Flat File

○ Flat File with Control Record

Use this page to associate a rule to a message and characterize the rule.

The page elements and controls that you use on the Batch Publish Rules page are described elsewhere in this documentation.

See Common Elements Used in The Publish Utility.

Record Mapping Page

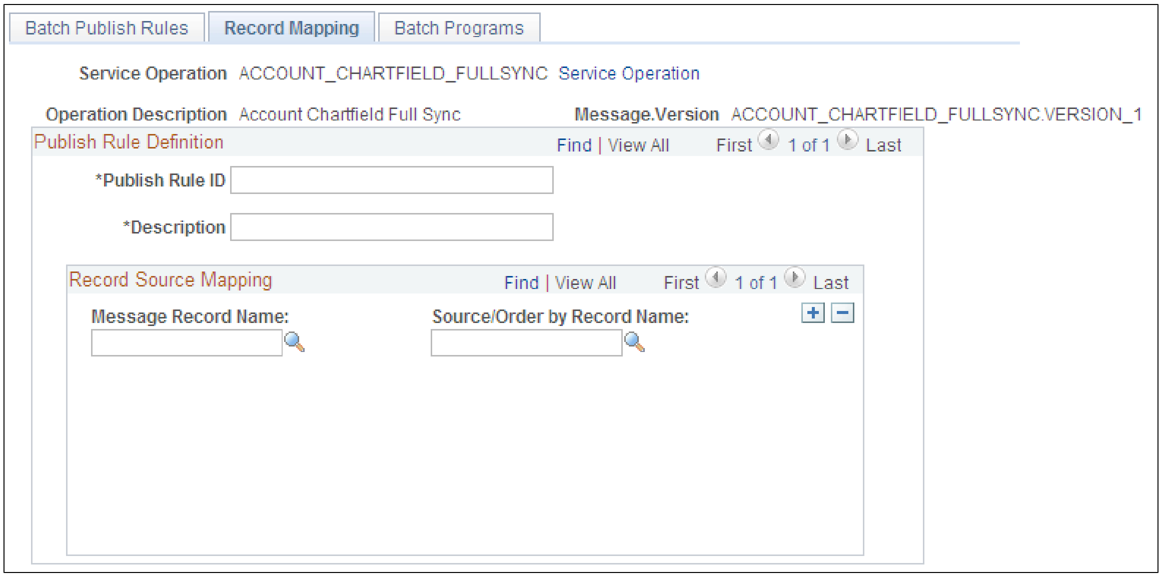
Use the Record Mapping page (EOIU_SORECMAP) to map a message record to another record.

Navigation

Enterprise Components, Integration Definitions, Batch Publish Rules, Record Mapping

Image: Record Mapping page

This example illustrates the fields and controls on the Record Mapping page. You can find definitions for the fields and controls later on this page.



Service Operation ACCOUNT_CHARTFIELD_FULLSYNC

Operation Description Account Chartfield Full Sync

Message.Version ACCOUNT_CHARTFIELD_FULLSYNC.VERSION_1

Use this page to map a message record to another record.

Note: When running a batch publish rule, the Publish utility runs cleanup logic, which either updates fields or deletes rows in the source tables. If the source table is a view that contains a join, then the option to delete published rows fails.

Message Record Name Enter the name of the record in the message that you want to map to another record.

Source/Order By Record Name Enter the record name that the Publish utility uses to select data.

This page specifies the source data for a record in a message. It works in the same manner and accomplishes the same purpose as the Record Mapping page for a full table publish.

Batch Programs Page

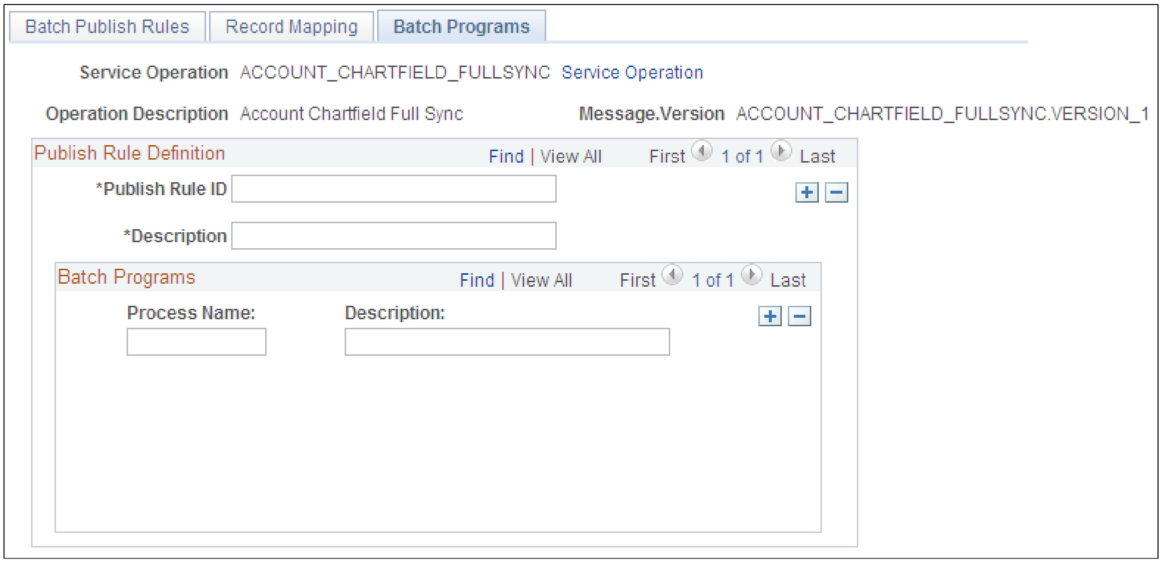
Use the Batch Programs page (EOIU_SOBATPGM) to assign an application program (PROCESS_NAME) to the publish rule.

Navigation

Enterprise Components, Integration Definitions, Batch Publish Rules, Batch Programs

Image: Batch Programs page

This example illustrates the fields and controls on the Batch Programs page. You can find definitions for the fields and controls later on this page.



Service Operation ACCOUNT_CHARTFIELD_FULLSYNC Service Operation

Operation Description Account Chartfield Full Sync

Message.Version ACCOUNT_CHARTFIELD_FULLSYNC.VERSION_1

Publish Rule Definition

*Publish Rule ID

*Description

Batch Programs

Process Name:

Description:

Use this page to assign an application program (PROCESS_NAME) to the publish rule.

Process Name

Enter the name of the COBOL, Structured Query Report (SQR), or Application Engine program that is marking the records to be published.

The Publish utility initially receives the process name from the batch parameter record that is created by the application program. The program then retrieves and processes each publish rule for the application process name. The process name can be any 12-character string, as long as it matches what the application program inserts into the batch parameter record.

If you select a flat file format, the Publish utility does not create a header or trailer message, and the utility ignores any chunking rules. Instead, the utility creates a single flat file.

Performing Pre-Processing and Post-Processing Tasks

This section discusses how to:

- Set general pre-processing and post-processing options.
- Code pre-processing and post-processing tasks.

Setting General Pre-Processing and Post-Processing Options

To perform pre-processing and post-processing tasks, you must select the appropriate Message Header or Message Trailer option on the Full Table Publish Rules page or on the Batch Publish Rules page.

To set general pre-processing and post-processing options:

1. Access the Full Table Publish Rules page or the Batch Publish Rules page.

    - To access the Full Table Publish Rules pages, select Enterprise Components, Integration Definitions, Full Table Publish Rules.

    - To access the Batch Publish Rules pages, select Enterprise Components, Integration Definitions, Batch Publish Rules.

2. In the Message Options section, select the appropriate options:

    Message Header Select this option to perform pre-processing tasks on a service operation message.

    Message Trailer Select this option to perform post-processing tasks on a service operation message.

3. Click the Save button.

Coding Pre-Processing and Post-Processing Tasks

The type of processing that you are performing determines the location in the message where you insert the pre-processing and post-processing logic.

Using PeopleSoft-Delivered Functions for Pre-Processing and Post-Processing

Consider using the following PeopleSoft-delivered functions in pre-processing logic:

Delete_Existing_Data Function The Delete_Existing_Data function loops through the default message definition records and deletes all the rows from each record. It also deletes all the rows from the related language record.

Subscribe_FullReplication Function The Subscribe_FullReplication function is called from subscription processes that do full replication. This function checks the first instance of PSCAMA.BATREPFIRSTMSG and if this flag is set to yes, it invokes delete processing that deletes all the data from every record in the message as well as related language records. It then invokes the same Proc_Sub_Rowset processing that incremental replication processing uses. It loops through the message hierarchy and inserts the data into the appropriate tables based on PSCAMA.AUDIT_ACTION and the message definition records.

Coding Pre-Processing and Post-Processing Tasks

The type of processing that you are performing, sequential or parallel, determines the location in the message where you insert pre-processing and post-processing logic.

The following code example shows sample pre-processing and post-processing code. Section numbers have been entered as remarks to show the location to insert code based on the processing that you are performing.

Processing	***Task***	***Section***
Parallel	Pre-processing	Section 1
Sequential	Pre-processing	Section 2
Sequential	Post-processing	Section 3
Parallel	Post-processing	Section 4

Important! If you change processing modes from sequential processing to parallel processing or vice versa, you must move any pre-processing and post-processing code to the appropriate section in the message, as described in the table.

```
If &ParallelFS Then
   ...
      If &Pre_Process_Flg = "Y" Then

         rem ****** Begin Pre-Processing Logic for parallel-enabled FullSync *****;
         Delete_Existing_Data(&MSG);


         /************************************************************************/
         /*                                                                      */
         /*                            Section 1                                 */
         /*                                                                      */
         /************************************************************************/

         rem ****** End Pre-Processing Logic for parallel-enabled FullSync ******;

      End-If;


End-If;

&MSG_ROWSET = &MSG.GetRowset();
&MSG_LANG_CD = &MSG_ROWSET(1).PSCAMA.LANGUAGE_CD.Value;
&MSG_BASE_LANG_CD = &MSG_ROWSET(1).PSCAMA.BASE_LANGUAGE_CD.Value;
&FULL_INCR = "FULL";

Evaluate &MSG_ROWSET(1).PSCAMA.MSG_SEQ_FLG.Value
When "H"
   If Not &ParallelFS Then

   rem ****** Begin Pre-Processing Logic for nonparallel-enabled FullSync ******;
   Delete_Existing_Data(&MSG);


    /************************************************************************/
    /*                                                                      */
    /*                                Section 2                             */
    /*                                                                      */
    /************************************************************************/
   rem ****** End Pre-Processing Logic for nonparallel-enabled FullSync ******;

   End-If;
   Break;
When "T"
   If Not &ParallelFS Then


    /*************************************************************************/
    /*                                                                       */
    /*                                Section 3                              */
    /*                                                                       */
    /*************************************************************************/


   rem ** Insert Post-Processing Logic Here for nonparallel-enabled FullSync **;

   End-If;
   Break;
When-Other
   Proc_Sub_Rowset(&MSG_ROWSET);
   rem ****** put additional process like editing here ********;

If &ParallelFS Then
    ...

   If &Post_Process_Flg = "Y" Then


    /*************************************************************************/
    /*                                                                       */
    /*                                Section 4                              */
    /*                                                                       */
    /*************************************************************************/


   rem *** Insert Post-Processing Logic Here for parallel-enabled FullSync ***;

   End-If;
   ...
   Break;

End-Evaluate;
```

Setting Up Message Chunking

This section provides an overview of message chunking and discusses how to:

- Identify when to use chunking.
- Select chunking fields.
- Create chunking rules.
- Define the chunking rule description.
- Maintain chunking data for business units.
- Maintain chunking data for setIDs.
- Maintain chunking data for record groups.
- Add nodes to existing chunking rules.

- Assign business units or setIDs to a chunking rule.
- Assign chunking rules to a business unit.
- Assign chunking rules to a setID.

Pages Used to Set Up Message Chunking

Page Name	***Definition Name***	***Navigation***	***Usage***
Chunking Rule	EO_CHUNKRULE	Enterprise Components, Integration Definitions, Map Chunking Rules, Define Chunking Rules	Define a chunking rule description.
BusUnit Mapping (business unit mapping)	EO_CHUNKBU	Enterprise Components, Integration Definitions, Map Chunking Rules, Business Units	Maintain chunking data for business units.
SetId Mapping	EO_CHUNKSETID	Enterprise Components, Integration Definitions, Map Chunking Rules, SetIds	Maintain chunking data for setIDs.
Eo Recgrp (Enterprise Component Record Group)	EO_RECGRP	Enterprise Components, Integration Definitions, Map Chunking Rules, Record Group	Maintain chunking data for record groups.
Add Nodes to Chunk Rule	EO_ADNODECHUNK_PNL	Enterprise Components, Integration Definitions, Map Chunking Rules, Node to ChunkRule	Add nodes to existing chunking rules.
Quick Map	EO_ADDSIDNODE_PNL	Click the Add button on the Add Notes to Chunk Rule page. This button is available after you have added nodes to the chunking rule.	Assign business units to a chunking rule.
Map Business Unit	EO_ADDNODEBU_PNL	Enterprise Components, Integration Definitions, Map Chunking Rules, ChunkRule/ Node to BU	Assign chunking rules to a business unit.
Map Set IDs	EO_ADDNODESID_PNL	Enterprise Components, Integration Definitions, Map Chunking Rules, ChunkRule/ Node to Setid	Assign chunking rules to a setID.

Understanding Message Chunking

If you publish to multiple nodes, you might want the messages to be routed based on a specific field. Chunking rules direct the message. You can use the same chunking fields for breaking a large message into smaller messages, as well as for associating messages with a node (based on those same fields). You set up this kind of relationship between nodes and the fields used to break the message apart (break fields) by using chunking message pages.

You can, depending on some XML content-based logic, use message chunking to route and deliver groups of transactions to different third-party nodes.

For example, consider purchase orders. If you run the full batch publish, each third-party node receives an XML message containing all purchase orders that are dispatched, regardless of whether any purchase orders are intended for that particular customer. With chunking, however, you can set up a chunking rule to chunk a message by customer ID. This creates an XML message for each customer that contains only purchase orders intended for that particular customer.

Note: Do not confuse message chunking with queue partitioning. You use queue partitioning to partition a queue by a level zero key field. If a field exists on level zero of the record in the message by which you can uniquely distinguish and group transactions to be processed in parallel, partitioning the message by this field increases performance. Without partitioning, a PeopleSoft subscribing system must process each incoming service operation and its associated message in the order in which the service operation is received according to the Publication ID (PUBID) field.

Identifying When to Use Chunking

Use chunking when:

- The message data is large, and the subscriber is consistently interested only in part of the data.
- Subscribers can more efficiently process the message data by chunking messages.
- You have trading-partner-specific content and legally do not want data to be shared among vendors.

You can chunk messages by:

- Locations and inventory shipments by business unit.
- Customers by setID.
- Employees by department or company.
- Sales order acknowledgements by setID and customer ID.
- Purchase orders and purchase order changes by setID and vendor ID.

Selecting Chunking Fields

To maximize performance and prevent application developers from maintaining complicated views of the data, create staging or temporary tables that contain the chunking fields as the highest order key fields.

Chunking fields can affect performance and alter the options that are available to the Publish utility. The Publish utility creates SQL for each table that is defined in the message object. Tables that are defined in the message can be mapped to an alternative source table or viewed on the Record Mapping page under the Publish Rule Definition page.

The source table (or view) serves two purposes:

- It enables the data that must be published to come from a source other than the table that is defined in the message.
- It enables the data to be ordered so that the Publish utility can process rows in the correct sequence.

Because the SQL is run only once and includes a subquery against the values in the chunking table, you must define all chunking fields in every table that is used to retrieve data for the message. The SQL order-by clause is set according to the key fields that are defined in the table. The result is that chunking fields must be defined as key fields for the Publish utility to work.

Example of Generated SQL for Chunking

This is an example of SQL that is generated for chunking:

```
Select * from PS_INV_ITEMS A
where EXISTS (Select 'Y' from PS_EO_SETID_EOC B
where B.CHUNK_RULE_ID = 'SETID'
and B.EFFDT = '20000201'
and A.SETID = B.SETID)
order by A.SETID, INV_ITEM_ID, EFFDT
```

The field that you select to chunk on determines the view table that you must create:

Chunking Field Attribute	*Corresponding View Table*
Chunking field is a key field in level zero table.	By rule, the chunking fields are also key fields in the child tables. The key fields of a parent table must be key fields in the child table and in the same order. If the chunking fields are not the highest order key fields, create a view that consists of all fields in the source table, with the chunking fields as the highest order key fields, followed by the remaining key fields from the source table. Then map this view table to the source table on the Record Mapping page under the Publish Rule Definition page.
Chunking field is not a key field.	Create a view of the source data that consists of all fields in the source table, with the chunking fields as the highest order key fields, followed by the rest of the key fields from the source table. Then map this view table to the source table on the Record Mapping page under the Publish Rule Definition page.
Chunking field is not in a source table.	Create a view table that joins the source table to an existing table that contains the chunking fields. This view must consist of all fields in the source table and the chunking fields from the joined table. The chunking fields are the highest order key fields, followed by the rest of the key fields from the source table. Then map the view table to the source table on the Record Mapping page under the Publish Rule Definition page.

Example

The following sample SQL code creates a view table that joins the PS_OMEC_CP_OPT_DET source table to an existing PS_OMEC_HDR_OUT table that contains the chunking fields. The

B.SETID_CUSTOMER and the B.CUST_ID chunking fields are the highest order key fields from the joined table (PS_OMEC_HDR_OUT), followed by the rest of the key fields from the source table (PS_OMEC_CP_OPT_DET).

```
SELECT B.SETID_CUSTOMER
, B.CUST_ID
, A.BUSINESS_UNIT
, A.ORDER_NO
, A.ORDER_INT_LINE_NO
, A.CP_MODE
, A.CP_COMP_SEQ
, A.OPTION_NAME
, A.OPTION_VALUE
, A.OPTION_DESC
, A.VAR_TYPE
, A.VAR_LENGTH
, A.VAR_DECIMAL
, A.PROCESS_INSTANCE
, A.AUDIT_ACTN
, A.IN_PROCESS_FLG
FROM PS_OMEC_CP_OPT_DET A
, PS_OMEC_HDR_OUT B
WHERE A.BUSINESS_UNIT = B.BUSINESS_UNIT
AND A.ORDER_NO = B.ORDER_NO
```

Creating Chunking Rules

The chunking rule consists of four tables:

Level	***Table***	***Description***
Level 0	EO_CHUNKRULE	A system table delivered with live data.
Level 1	EO_CHUNKEFFDT	A system table delivered with live data. When a chunking rule is saved, a row is added to this table with the effective date (EFFDT) field automatically populated from the current date and the effective status set to *Active.*
Level 2	EO_CHUNKNODE	This is not a system table and is delivered empty.
Level 3	NAME_EOC	A user-defined chunking table.

Note: All user-defined chunking table names must end in *_EOC.*

Oracle provides three standard tables: EO_BUSUNIT_EOC for business unit values, EO_SETID_EOC for setID values, and EO_RECGRP_EOC for record group values. The different types of user-defined chunking tables are:

Table Type	***Description***
Derived Table	Contains only the chunking fields. This table can be used by the Publish utility to chunk the message whenever the value of the chunking field changes. In derived tables, no relationship exists between the value of the chunking fields and message node names that are used to route the message. OnRoute PeopleCode needs hard-coded routing logic or additional tables to route the message to the appropriate nodes.
SQL Tables	Contains the following fields: • CHUNK_RULE_ID • EFFDT • MSGNODENAME • Chunking fields This table limits the published data to the values of the chunking fields in the chunking table and contains the message node name that is used to route the message.
Alternate Chunking Tables	Enables reuse of existing chunking tables. This table must end in *_EOV*.

Chunking Rule Page

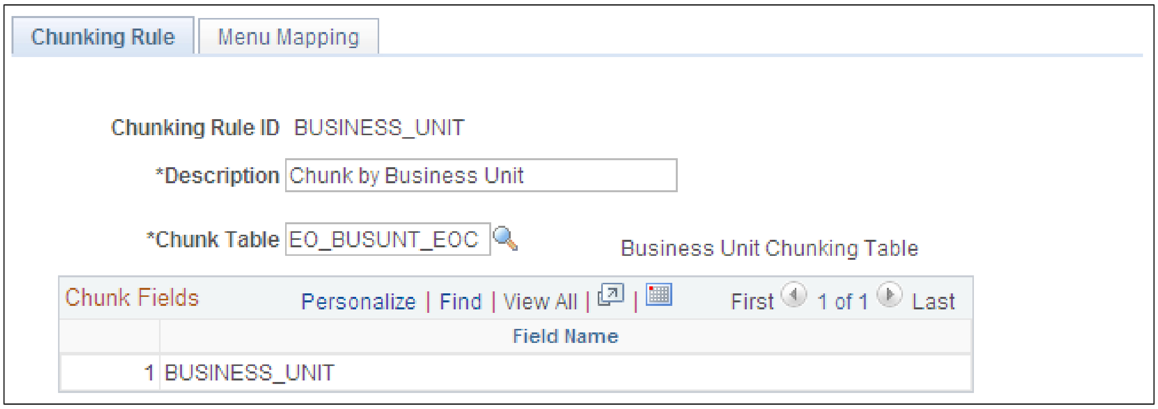
Use the Chunking Rule page (EO_CHUNKRULE) to define a chunking rule description.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, Define Chunking Rules

Image: Chunking Rule page

This example illustrates the fields and controls on the Chunking Rule page. You can find definitions for the fields and controls later on this page.



Chunking Rule ID BUSINESS_UNIT

*Description Chunk by Business Unit

*Chunk Table EO_BUSUNT_EOC Business Unit Chunking Table

	Field Name
1	BUSINESS_UNIT

Use this page to define a chunking rule description.

Chunk fields are the chunking fields that are defined in the Chunk Table field.

The chunking fields appear in this scroll area to verify that the correct chunking table was entered. Oracle provides chunking rules for business unit and setID for all application databases. Adding a new chunking rule inserts a row into the EO_CHUNKEFFDT table, with a default effective date (EFFDT) of the current date.

BusUnit Mapping Page

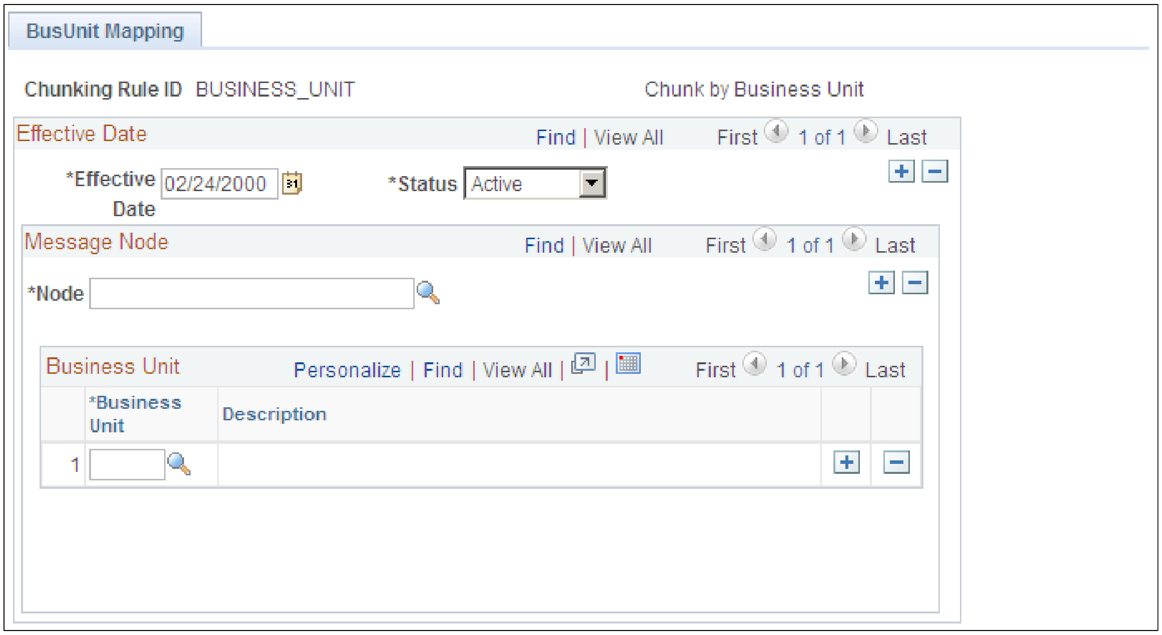
Use the BusUnit Mapping (business unit mapping) page (EO_CHUNKBU) to maintain chunking data for business units.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, Business Units

Image: BusUnit Mapping page

This example illustrates the fields and controls on the BusUnit Mapping page. You can find definitions for the fields and controls later on this page.



BusUnit Mapping

Chunking Rule ID BUSINESS_UNIT Chunk by Business Unit

Effective Date Find | View All First 1 of 1 Last

*Effective Date 02/24/2000 *Status Active

Message Node Find | View All First 1 of 1 Last

*Node

Business Unit Personalize | Find | View All | First 1 of 1 Last

	*Business Unit	Description
1		

Use this page to maintain chunking data for business units.

All four levels of the chunking rule tables appear.

Oracle provides chunking tables for business unit, setID and record group that are maintained by a series of components (such as components that are created for maintaining the business unit chunking table).

You can use each component to update the underlying relationship between the business unit and the subscribing nodes. You can maintain the data either by business unit or by node, individually or as a group, to reduce the amount of entry work.

SetId Mapping Page

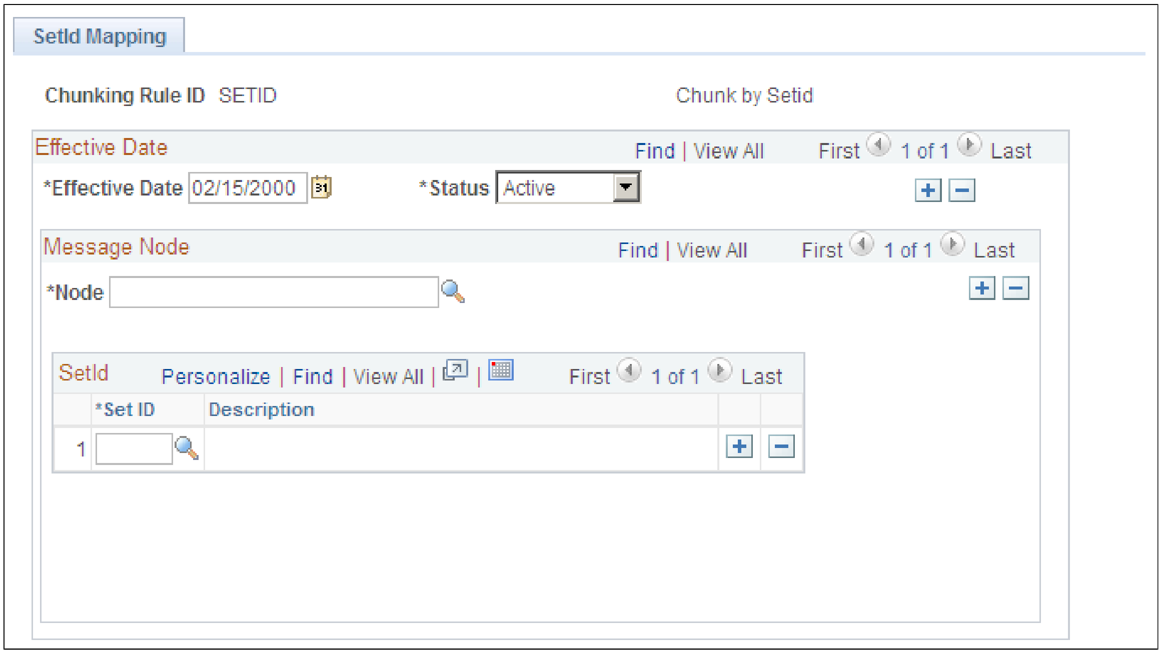
Use the SetId Mapping page (EO_CHUNKSETID) to maintain chunking data for setIDs.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, SetIds

Image: SetId Mapping page

This example illustrates the fields and controls on the SetId Mapping page. You can find definitions for the fields and controls later on this page.



SetId Mapping
Chunking Rule ID SETID
Chunk by Setid
Effective Date
Find | View All First 1 of 1 Last
*Effective Date 02/15/2000
*Status Active
Message Node
Find | View All First 1 of 1 Last
*Node
SetId Personalize | Find | View All | | First 1 of 1 Last
*Set ID
Description
1

Use this page to maintain chunking data for setIDs.

All four levels of the chunking rule tables appear.

Oracle provides chunking tables for business unit, setID and record group that are maintained by a series of components (such as components that are created for maintaining the business unit chunking table).

You can use each component to update the underlying relationship between the business unit and the subscribing nodes. You can maintain the data either by business unit or by node, individually or in a group, to reduce the amount of entry work.

Eo Recgrp Page

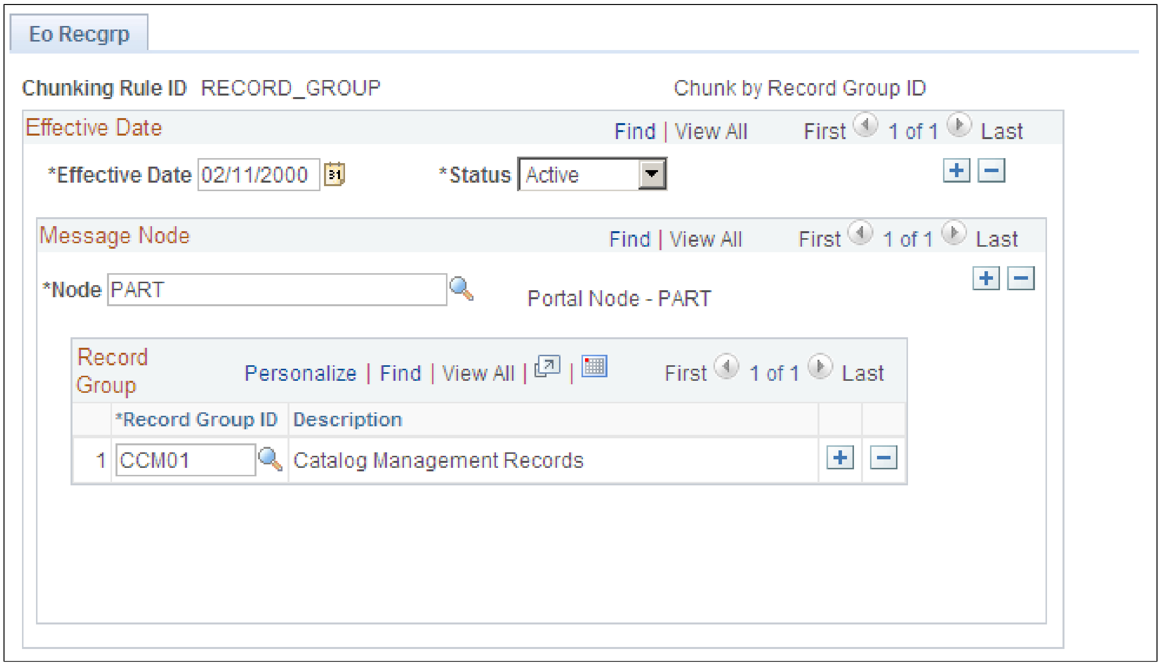
Use the Eo Recgrp (Enterprise Component Record Group) page (EO_RECGRP) to maintain chunking data for record groups.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, Record Group

Image: Eo Recgrp page

This example illustrates the fields and controls on the Eo Recgrp page. You can find definitions for the fields and controls later on this page.



Chunking Rule ID RECORD_GROUP Chunk by Record Group ID

*Effective Date 02/11/2000 *Status Active

*Node PART Portal Node - PART

	*Record Group ID	Description
1	CCM01	Catalog Management Records

Use this page to maintain chunking data for record groups.

All four levels of the chunking rule tables appear.

Oracle provides chunking tables for business unit, setID and record group that are maintained by a series of components (such as components that are created for maintaining the business unit chunking table).

You can use each component to update the underlying relationship between the business unit and the subscribing nodes. You can maintain the data either by business unit or by node, individually or in a group, to reduce the amount of entry work.

Add Nodes to Chunk Rule Page

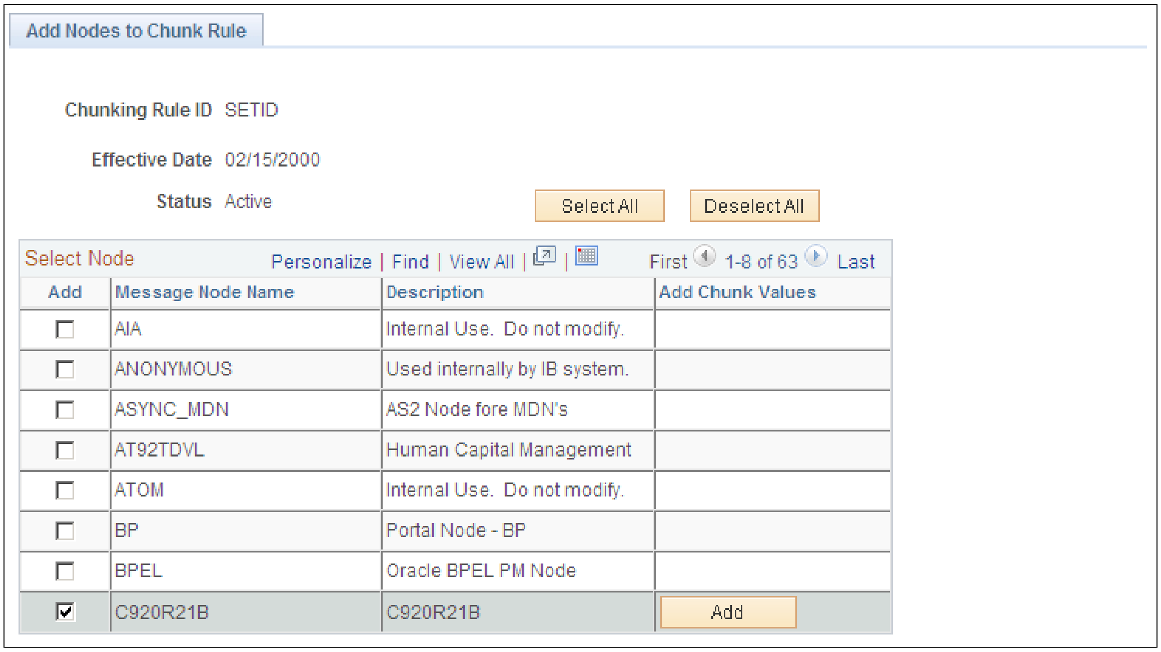
Use the Add Nodes to Chunk Rule page (EO_ADNODECHUNK_PNL) to add nodes to existing chunking rules.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, Node to ChunkRule

Image: Add Nodes to Chunk Rule page

This example illustrates the fields and controls on the Add Nodes to Chunk Rule page. You can find definitions for the fields and controls later on this page.



Chunking Rule ID SETID

Effective Date 02/15/2000

Status Active

Select Node

Add	Message Node Name	Description	Add Chunk Values
□	AIA	Internal Use. Do not modify.	
□	ANONYMOUS	Used internally by IB system.	
□	ASYNC_MDN	AS2 Node fore MDN's	
□	AT92TDVL	Human Capital Management	
□	ATOM	Internal Use. Do not modify.	
□	BP	Portal Node - BP	
□	BPEL	Oracle BPEL PM Node	
☑	C920R21B	C920R21B	Add

Use this page to add nodes to existing chunking rules.

To add nodes to an existing chunking rule:

1. Select the check box in the Add column of the nodes that you defined earlier.
2. Click the Save button to display the Add Chunk Values column.

   When you select a node and then click Save, the ADD button in the Add Chunk Values column appears.
3. Click the Add button in the Add Chunk Values column for the nodes that you want to add.

   The Quick Map page appears.

Quick Map Page

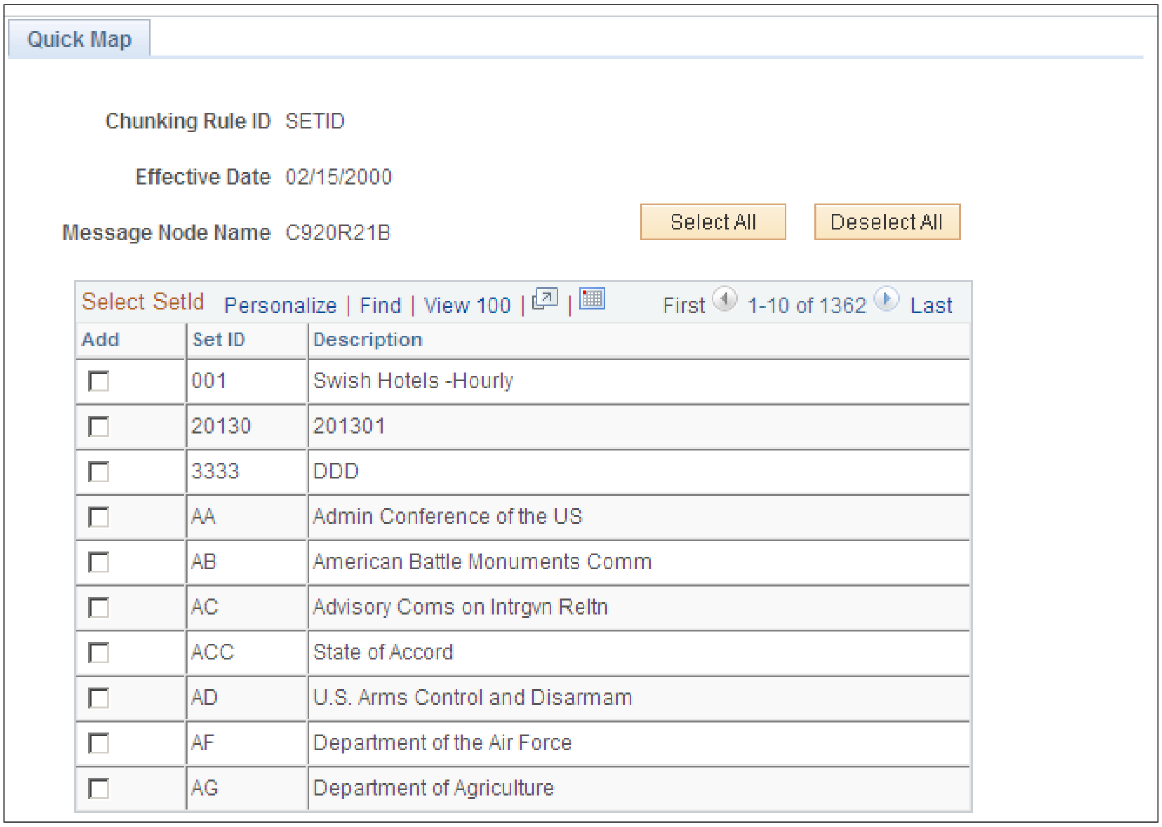
Use the Quick Map page (EO_ADDSIDNODE_PNL) to assign business units to a chunking rule.

Navigation

Click the Add button on the Add Notes to Chunk Rule page. This button is available after you have added nodes to the chunking rule.

Image: Quick Map page

This example illustrates the fields and controls on the Quick Map page. You can find definitions for the fields and controls later on this page.



Chunking Rule ID	SETID
Effective Date	02/15/2000
Message Node Name	C920R21B

Add	Set ID	Description
□	001	Swish Hotels -Hourly
□	20130	201301
□	3333	DDD
□	AA	Admin Conference of the US
□	AB	American Battle Monuments Comm
□	AC	Advisory Coms on Intrgvn Reltn
□	ACC	State of Accord
□	AD	U.S. Arms Control and Disarmam
□	AF	Department of the Air Force
□	AG	Department of Agriculture

Use this page to a Assign business units to a chunking rule.

If you previously accessed business unit chunking rules, you can add business units to a chunking rule. If you previously accessed setIDs, you can add setIDs to a chunking rule.

Note: You cannot access the Quick Map page without first using either the BusUnit Mapping page or the SetId Mapping page to add an effective-dated node to the chunking rule ID.

Select All and Deselect All Click to add or remove all business units that are assigned to the node. Add check boxes are selected for business units that are assigned to the node.

Map Business Unit Page

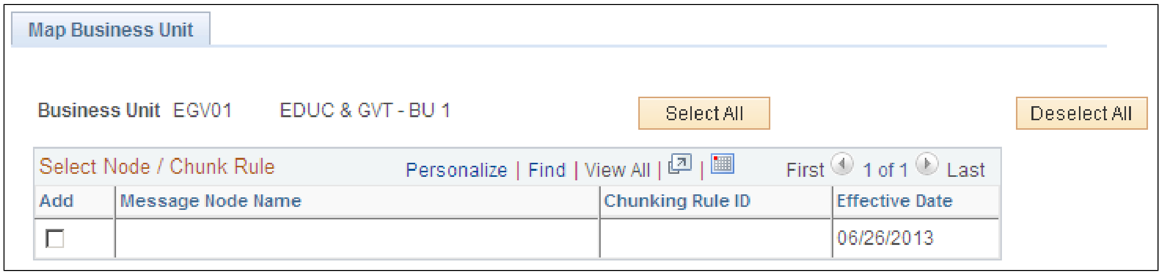
Use the Map Business Unit page (EO_ADDNODEBU_PNL) to assign chunking rules to a business unit.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, ChunkRule/Node to BU

Image: Map Business Unit page

This example illustrates the fields and controls on the Map Business Unit page. You can find definitions for the fields and controls later on this page.



Business Unit EGV01 EDUC & GVT - BU 1

Add	Message Node Name	Chunking Rule ID	Effective Date
□			06/26/2013

Use this page to assign chunking rules to a business unit.

Select All and Deselect All

Click to add or remove all nodes that are assigned to the business unit. Add check boxes are selected for nodes that are assigned to the business units.

Map Set IDs Page

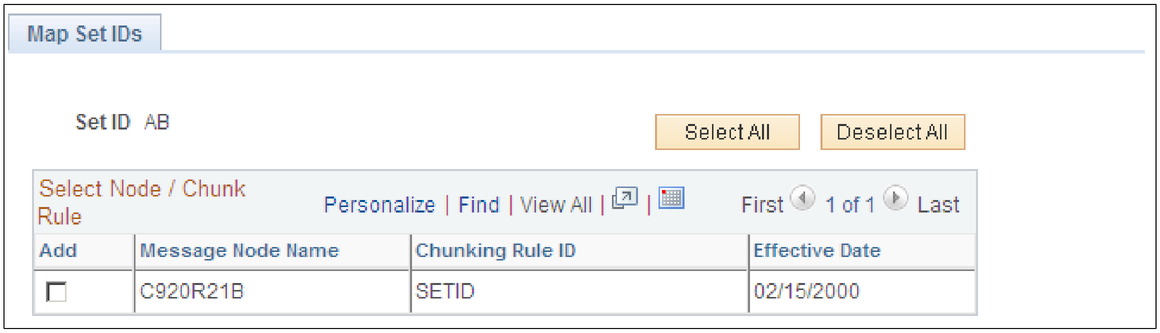
Use the Map Set IDs page (EO_ADDNODESID_PNL) to assign chunking rules to a setID.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, ChunkRule/Node to Setid

Image: Map Set IDs page

This example illustrates the fields and controls on the Map Set IDs page. You can find definitions for the fields and controls later on this page.



Set ID AB

Add	Message Node Name	Chunking Rule ID	Effective Date
□	C920R21B	SETID	02/15/2000

Use this page to assign chunking rules to a setID.

Select All and Deselect All

Click to add or remove all message nodes that are assigned to the setID. Add check boxes are selected for message nodes that are assigned to the setIDs.

Creating Custom Chunking Tables

This section discusses how to:

- Create a custom chunking table.
- Create a view for the component search record.
- Create maintenance pages.
- Create a component.
- Create routing PeopleCode.

Creating a Custom Chunking Table

To create a custom chunking table:

1. Select File, Open in PeopleSoft Application Designer.
2. Select *Record* in the Definition drop-down menu.
3. Open the EO_BUSUNT_EOC record.
4. Save the record as YOUR_TABLE_EOC.
5. Remove the BUSINESS_UNIT field.
6. Insert the custom chunking fields at the bottom of the record.
7. Select File, Save.
8. Build the SQL table.

Creating a View for the Component Search Record

To create a view for the component search record:

1. Select File, Open in PeopleSoft Application Designer.
2. Select *Record* in the Definition drop-down menu.
3. Open the EO_CHUNKBU_VW record.
4. Save the record as YOUR_TABLE_VW.
5. Select the Record Type tab.
6. Open the SQL editor.
7. Modify the Where clause.

   Change `WHERE RECNAME_CHUNK=EO_BUSUNT_EOC` to `WHERE RECNAME_CHUNK=YOUR_TABLE_EOC`.

8. Select File, Save.

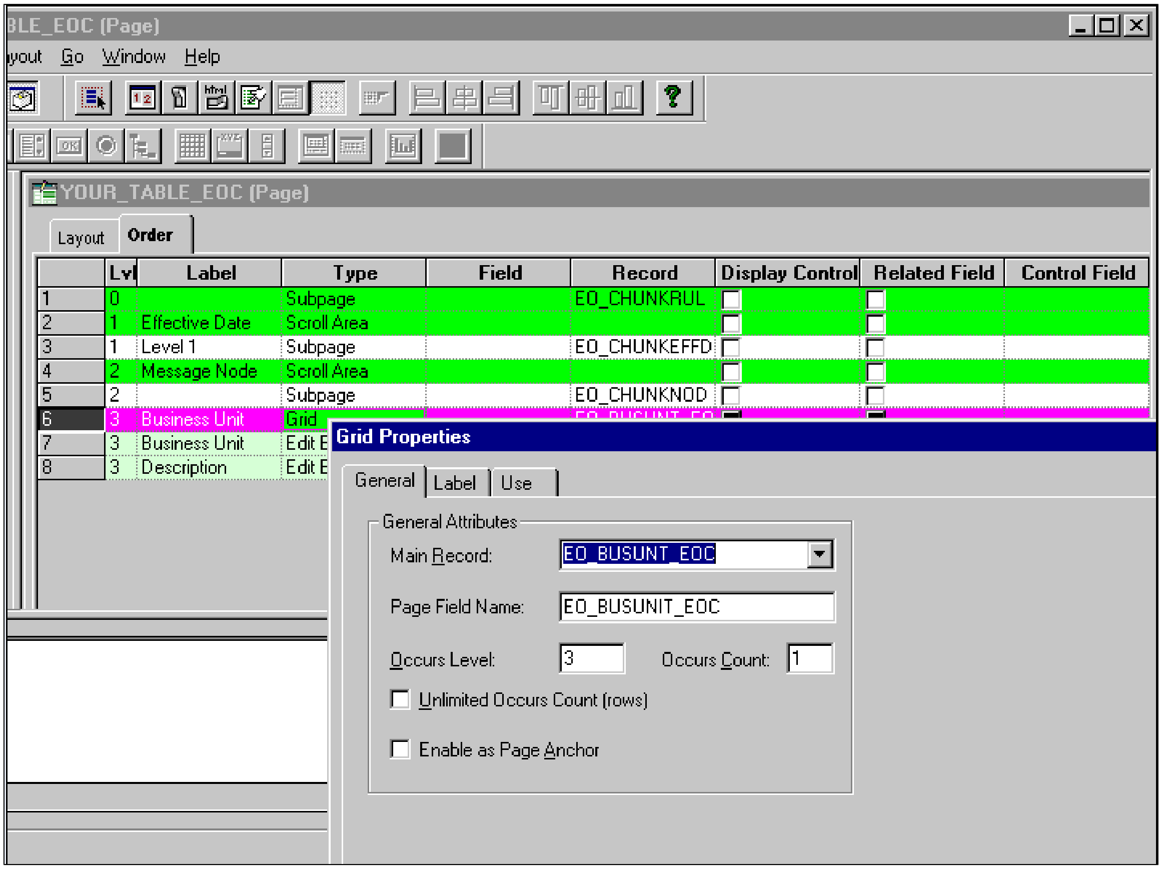
9. Build the SQL view.

Creating Maintenance Pages

You can also create maintenance pages.

Image: Grid Properties dialog box

This example illustrates the fields and controls on the Grid Properties dialog box. You can find definitions for the fields and controls later on this page.



	Lvl	Label	Type	Field	Record	Display Control	Related Field	Control Field
1	0		Subpage		EO_CHUNKRUL	□	□	
2	1	Effective Date	Scroll Area			□	□	
3	1	Level 1	Subpage		EO_CHUNKEFFD	□	□	
4	2	Message Node	Scroll Area			□	□	
5	2		Subpage		EO_CHUNKNOD	□	□	
6	3	Business Unit	Grid		EO_BUSUNIT_EO			
7	3	Business Unit	Edit E					
8	3	Description	Edit E					

Main Record: EO_BUSUNT_EOC

Page Field Name: EO_BUSUNIT_EOC

Occurs Level: 3 Occurs Count: 1

□ Unlimited Occurs Count (rows)

□ Enable as Page Anchor

To create maintenance pages:

1. Select File, Open in PeopleSoft Application Designer.

2. Select *Page* in the Definition drop-down menu.

3. Open the EO_CHUNKBU page.

4. Save the page as *YOUR_PAGE.*

5. Select the Order tab.

6. In the Type column, double-click Grid to open the Grid Properties dialog box.

7. Change the value in the Main Record and Page Field Name fields to *YOUR_TABLE_EOC.*

8. Click OK.

9. Delete the business unit and description columns.

10. Add chunking fields from YOUR_TABLE_EOC.

11. Select File, Save.

Creating a Component

To create a component:

1. Select File, New in PeopleSoft Application Designer.

2. Select *Component* in the New Definition box.

3. Select Insert, Page Into Component.

4. Enter *YOUR_PAGE.*

5. Click Close.

6. Select File, Definition Properties

7. Select the Use tab to edit component properties:

    a. In the Search record field, enter *YOUR_TABLE_VW.*

    b. Select the Update/Display, Update/Display All, and Correction check boxes.

8. Click OK.

9. Select File, Save As, and save the page group as *YOUR_COMPONENT.*

10. Add YOUR_PAGE_GROUP to YOUR_MENU that is used by your application.

Creating Routing PeopleCode

OnRouteSend (and OnRouteReceive) are PeopleCode methods that are tied to the message for routing, based on the message contents. If you want the contents of the message (such as a message chunking field value) to determine the subscribing nodes that should receive the message, OnRouteReceive PeopleCode must contain the logic to examine the message and return a list of subscribing nodes.

The OnRouteSend and OnRouteReceive methods are contained in the IRouter application class. The IRouter application class is located in PeopleSoft Application Designer in the Integration subpackage of the PS_PT application package.

PeopleCode functions provided by common components, GetNodes and RtnNodes, work with any message and chunking rule. For a given message, these nodes select the chunking rule for the publish rule that is assigned to the message.

The functions:

- Build SQL based on the chunking fields as defined in the chunking table.

- Extract chunking field values from the message.
- Run the associated SQL.
- Compare the array of nodes returned to the application server against the list of nodes for the message channel.
- Create a publish contract for nodes in both arrays.

  You can override the publish rule from the message, specifying an optional parameter when calling the functions.
- Return an array of nodes that is based on the nodes that are assigned to the message channel if the publish rule is invalid or does not contain a chunking rule.

  Returning an array of nodes enables the functions to work regardless of whether chunking is set up for the publish rule.

To route any message that uses chunking, use generic PeopleCode functions.

These functions are called from routing PeopleCode:

GetNodes

Returns an array of nodes to the application server.

Use this function for integrations on PeopleTools 8.47 and earlier releases.

RtnNodes

Returns an array of nodes to the calling PeopleCode.

Use this function for integrations on PeopleTools 8.47 and earlier releases.

RtnMsgNodes

Returns an array of nodes of datatype Any to the calling PeopleCode.

Use this function for integrations on PeopleTools 8.48 and higher releases.

These functions are internal functions:

FndNodes

Builds an array of nodes for the message.

GetPubRule

Selects the chunking rule for the publish rule.

GetChunkInfo

Selects the chunk table for the chunking rule.

BuildSQL

Builds SQL to select nodes from the chunking table for specific chunking field values from the message.

GetValue

Gets the chunking field values from the message.

HasNodes

Determines whether a chunking field is mapped to any nodes for a particular chunking rule.

The following code example shows the logic that you can add to SavePostChange PeopleCode for the Customer_General component to verify that the setID can publish the message by calling the HasNodes() function:

```
Declare Function HasNodes PeopleCode FUNCLIB_EOEIP.PUBLISH_ROUTE_PC
FieldFormula;
Local Message &MSG;
Local Rowset &RS0;
Local string &PublishRule;
&MSG = CreateMessage(MESSAGE.CUSTOMER_MSG);
/* Check if message is active */
If &MSG.IsActive Then
     &RS0 = GetLevel0();
     &PublishRule = "CUSTOMER_SYNC";
     /* Call function passing publish rule and rowset, which returns
     true if this setID can publish the message */
     If (HasNodes(&PublishRule, &RSO)) Then
          &RS0 = GetLevel0();
          &MSG.CopyRowsetDelta(&RS0);
          &MSG.Publish();
     End-If;
End-If;
```

The following code example shows the logic that you can add to service operation APC handler PeopleCode to chunk the message by nodes as defined in the chunking rules by calling the RtnMsgNodes function:

```
import PS_PT:Integration:IRouter;

class ChunkSetidByNode implements PS_PT:Integration:IRouter
   method RoutingHandler();
   property array of any destinationNodes;
   method OnRouteSend(&_MSG As Message) Returns integer;
   method OnError(&_MSG As Message);
end-class;

Declare Function RtnMsgNodes PeopleCode FUNCLIB_EOEIP.PUBLISH_ROUTE_PC
FieldFormula;

/* constructor */
method RoutingHandler
end-method;

method OnRouteSend
   /+ &_MSG as Message +/
   /+ Returns Integer +/
   /+ Extends/implements PS_PT:Integration:IRouter.OnRouteSend +/
   /* Variable Declaration */
   Local string &PublishRule;

   %This.destinationNodes = RtnMsgNodes(&PublishRule, &_MSG);

   If %This.destinationNodes.Len > 0 Then
      Return (%IntBroker_ROUTE_SOME);
   Else
      Return (%IntBroker_ROUTE_ALL);
   End-If;

end-method;

/** If an error occurs the OnError method if implemented will be automatically invo⇒
ked. The type of exception can be viewed by using the Message object to retrieve th⇒
e Exception object (&Message.IBException)
  * @param   MSG Message object containing the operation instance where the error o⇒
ccured while being routed
  */
method OnError
```

```
   /+ &_MSG as Message +/
   /+ Extends/implements PS_PT:Integration:IRouter.OnError +/
end-method;
```

Chapter 6

Using the Inbound Data Error Scan Utility

Understanding the Inbound Data Error Scan Utility

The Inbound Data Error Scan utility launches the PeopleSoft Application Engine program EO_WF_ERR that scans the PSAPMSGSUBCON and EO_EIP_CTL tables for data errors on inbound asynchronous integrations and notifies respective users if errors are encountered.

Related Links

Understanding the Error Handling Utility

Running the Inbound Data Error Scan Utility

This section discusses how to launch the Inbound Data Error Scan utility.

Page Used to Run the Inbound Data Error Scan Utility

Page Name	***Definition Name***	***Navigation***	***Usage***
Incoming Data Error Scan	EO_ERR_RUNCNTL	Enterprise Components, Integration Definitions, Initiate Processes, Inbound Data Error Scan	Start the EO_WF_ERR application engine program.

Incoming Data Error Scan Page

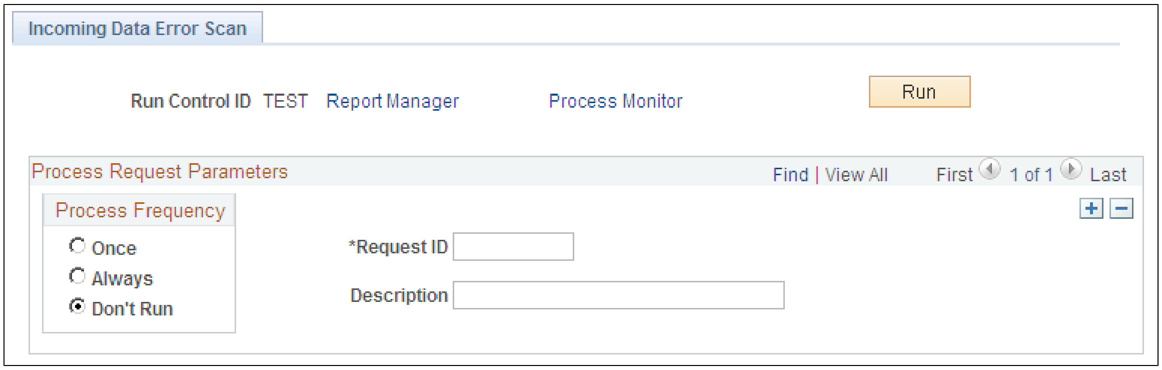
Use the Incoming Data Error Scan page (EO_ERR_RUNCNTL) to start the EO_WF_ERR application engine program.

Navigation

Enterprise Components, Integration Definitions, Initiate Processes, Inbound Data Error Scan

Image: Incoming Data Error Scan page

This example illustrates the fields and controls on the Incoming Data Error Scan page. You can find definitions for the fields and controls later on this page.



Run Control ID TEST

○ Once
○ Always
● Don't Run

*Request ID

Description

Request ID Enter the request ID of the integration on which to scan for errors.

Run Click to launch the EO_WF_ERR application engine program.

Chapter 7

Using the Effective Date Publish Utility

Understanding the Effective Date Publish Utility

The Effective Date Publish utility enables you to design processes to update external systems that process only current data and don't use or recognize effective dating.

When working with effective dating and effective date publishing, you need to understand the following terms:

Current Row

The current row is the first row of data with an effective date equal to or prior to the system date. Only one row can be the current row.

Future Rows

Future rows have effective dates later than the system date (usually the current date).

Historical Rows

Historical rows have effective dates prior to the current row.

Effective Date

An effective date is when a table row becomes effective, or the date that an action begins. The PeopleSoft system supports the concept of effective-dated rows.

Note: The EFFDT field is almost always a key. Specify the descending key attribute to display the row with the most recent effective date first.

Effective Dating

Automated effective dating saves changed data in a staging table for subsequent processing when the effective date becomes current. (Although data can be historical, current, or future, some third-party applications may support only current data. Thus, if a future-dated row is created within the PeopleSoft system, it must be delayed before transmission to the other system.)

Effective Sequence

An effective sequence serves two purposes:

- If EFFSEQ is a required field, it enables the entry of more than one row with the same effective date when paired with EFFDT. The system assigns a unique sequence number to each row that has the same effective date. It also enables the first EFFSEQ to be zero.
- If EFFSEQ is not a required field, it is not paired with EFFDT, has no special function, and can be used as a simple sequencing field.

Effective Status

Effective status enables the system to select the appropriate effective-dated rows, when combined with the effective date field.

Full Data Publish

The full data publish process seeds, or initially populates or repopulates, a copy of an entire table onto a remote database or legacy system. The entire contents of the table are published to all systems that require a copy of the table. Generally, full data replication occurs with setup tables (relatively static, low-volume tables that are keyed by setID) and occurs in an asynchronous manner.

When a full copy of the table exists on the external system, an incremental update provides a mechanism to keep the copy up-to-date with changes made on the master.

Incremental Publish

The incremental publish process sends a message that contains only the rows where the data has been modified, plus the corresponding anchoring parent and grandparent rows. When a particular transactional event occurs, an incremental update of the transaction data is sent to other systems to notify them of the changes.

Nodes

Each node represents a publishing or subscribing system of a service operation. For example, the PeopleSoft Human Resources and PeopleSoft Financials databases are each defined as a node, even if they are both on the same server.

Service Operation Queues

Service operation queues group messages and the nodes to which they are published, so that messages are published sequentially. Each message must belong to only one queue. Queues control the ordering of messages and define timeout parameters and error thresholds. Assign nodes to a queue when you define the message in the service operation.

Message Chunking

Chunking automatically breaks a message into several smaller messages based on the values in one or more of the fields in the level zero record. When publishing the entire contents of a table, you can use message chunking to publish only certain sets of data if, for example, a particular subscriber is interested in only a portion of the table.

Request ID

Use the request ID to specify multiple requirements within the same run control.

Run Control

You use run controls to produce full messages for objects at the same time. Run controls also associate publish rule definitions with the scheduled full publish process run. For example, you can set up a run control to publish both customer full messages and sales order full messages on a daily schedule.

Performing a Full Data Publish of Current Effective Data

This section discusses how to:

- Create effective-dated messages.
- Create a service operation.
- Define the node and target connector.
- Define chunking rules and ordering views.
- Create publish rule definitions.
- Create run controls for the Full Data Publish program.
- (Optional) Define message routing.

For full data messages that are intended for vendors who do not handle effective dating, use the Effective Date Publish utility and a current full message to publish only those rows that are currently active. Any future-dated rows are written to the delay table.

This section discusses the process involved in a full data publish of current effective data. It uses the CUSTOMER_FULLSYNC_EFF service operation as an example, but the methods and procedures that are described here apply to creating any effective-dated service operation that contains effective-dated messages.

Pages Used to Perform a Full Data Publish of Current Effective Data

Page Name	*Definition Name*	*Navigation*	*Usage*
Chunking Rule	EO_CHUNKRULE	Enterprise Components, Integration Definitions, Map Chunking Rules, Define Chunking Rules	Define the chunking rule description.
Full Data Publish	EO_FULLDATAPUB	Enterprise Components, Integration Definitions, Initiate Processes, Full Data Publish	Create the run control for the Full Data Publish utility. The run control associates publish rule definitions with the scheduled Full Publish process run. For example, you can set up a run control to publish both customer and sales order full messages at the end of each day.

Creating Effective-Dated Messages

The structure of the current full message must be a clone of the original FullSync message structure. However, you must map effective-dated records to a record view that selects only those rows that contain current data.

This example uses the message CUSTOMER_FULLSYNC. The current full message for customer data, CUSTOMER_FULLSYNC, uses the following views that are created as ordering view records.

Level	***Target Records***	***Ordering View Records***
Level 0	CUSTOMER	CUSTOMER
Level 1	CUST_ADDR_CNTCTC	CUST_ADDR_CNTCTC
Level 1	CUST_ADDR_SEQ	CUST_ADDR_SEQ
Level 2	CUST_ADDRESS (effective-dated)	CUST_ADDR_EF2VW (current effective-dated view)
Level 1	CUST_CNTCT_SEQ	CUST_CNTCT_SEQ
Level 2	CUST_CONTACT (effective-dated)	CUST_CNCT_EF2VW (current effective-dated view)
Level 3	CUST_CNTCT_CARD (effective-dated)	CUST_CARD_EF_VW (current effective-dated view)
Level 3	CUST_CNTCT_DOC (effective-dated)	CUST_DOC_EF_VW (current effective-dated view)
Level 3	CUST_CNTCT_PHN (effective-dated)	CUST_PHN_EF_VW (current effective-dated view)
Level 3	CUST_CNTCT_TYPE (effective-dated)	CUST_TYPE_EF_VW (current effective-dated view)

Creating the Service Operation

To complete a full data publish of effective dated data, you must create a service operation.

Note: The name of the service operation must end with *_EFF* suffix to be effective dated.

When you create a service operation:

- Specify the effective dated messages to publish.
- Specify a service operation queue for each message.
- Specify the directionality or routing of the integration.
- Define an OnRouteSend service operation handler, which contains the processing logic.

Defining the Node and Target Connector

Create a full message definition that contains the necessary records in the publishing system. You can also set up message routing by using OnRouteTo PeopleCode.

Note: Remember to insert the message version first, otherwise you can't add the tables that compose the message.

Begin by setting up the node and the transaction and connector details by using the Integration Profile setup function of PeopleSoft Integration Broker. To set up the node:

1. Create a node.

   See *PeopleTools: PeopleSoft Integration Broker Administration*, “Adding and Configuring Nodes”

2. Set up the connector.

   The default target is PSHTTP, but you can instead provide the HTTP address for another target.

   See *PeopleTools: PeopleSoft Integration Broker Administration*, “Managing Integration Gateways”

Chunking Rule Page

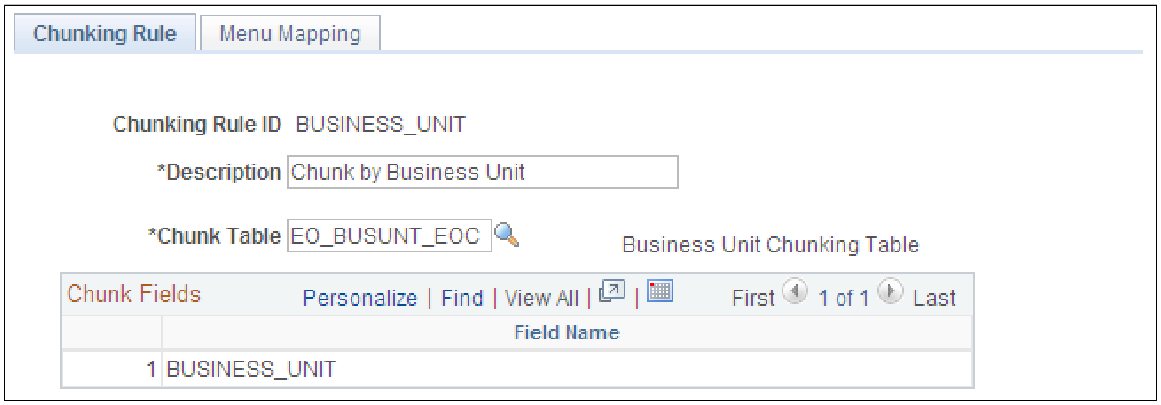
Use the Chunking Rule page (EO_CHUNKRULE) to define the chunking rule description.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, Define Chunking Rules

Image: Chunking Rule page

This example illustrates the fields and controls on the Chunking Rule page. You can find definitions for the fields and controls later on this page.



Chunking Rule ID BUSINESS_UNIT

*Description Chunk by Business Unit

*Chunk Table EO_BUSUNT_EOC Business Unit Chunking Table

	Field Name
1	BUSINESS_UNIT

In message chunking, all data within the message contains the same break field. For example, if the break field is business unit, all transactions in the message are for the same business unit.

To ensure that the message publishes when you use chunking rules:

1. Add subscribing nodes to the chunking node table.
2. In PeopleSoft Application Designer, add OnRouteSend PeopleCode to return a list of subscribing nodes.

Creating Publish Rule Definitions

For creating the current full message, the publish rule defines these options:

- Message header and trailer creation.
- Chunking rules.

- Ordering views.

Create publish rule definitions for each current full message definition. Observe these guidelines:

- Specify only target records that are effective-dated.
- Select only the current effective row to list the ordering view record that should be used as an override when the message is published.
- The Effective Date Publish utility makes a logic pass through the data for each publish rule definition.

  You can use this logic to order and chunk the data differently for each subscriber.

Note: When chunking a message, you must provide an ordering view for each record that includes the chunking fields. The fields in this view must appear in the same order as the primary keys, followed by any other keys that are needed for that record. If you override the normal key structure of the message records, you must provide the ordering views for each record to guarantee that the message is reconstructed with the correct chunking, parent, or child key relationships.

Full Data Publish Page

Use the Full Data Publish page (EO_FULLDATAPUB) to create the run control for the Full Data Publish utility.

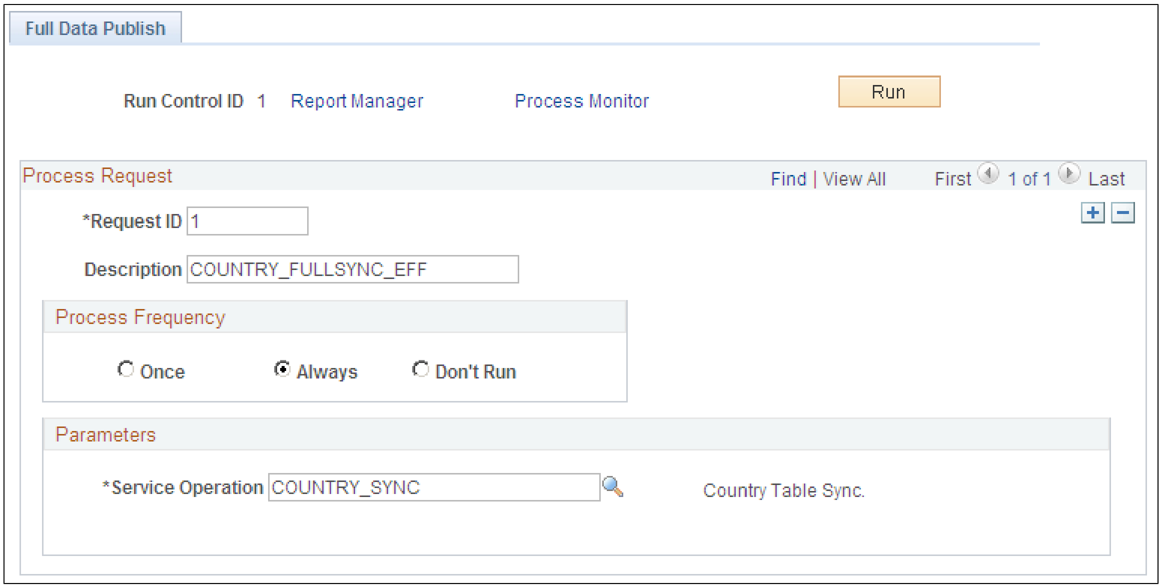
The run control associates publish rule definitions with the scheduled Full Publish process run. For example, you can set up a run control to publish both customer and sales order full messages at the end of each day.

Navigation

Enterprise Components, Integration Definitions, Initiate Processes, Full Data Publish

Image: Full Data Publish page

This example illustrates the fields and controls on the Full Data Publish page. You can find definitions for the fields and controls later on this page.



Run Control ID 1

*Request ID 1

Description COUNTRY_FULLSYNC_EFF

○ Once ● Always ○ Don't Run

*Service Operation COUNTRY_SYNC Country Table Sync.

Use this page to create the run control for the Full Data Publish utility.

Request ID

Enter request IDs to group the Description, Process Frequency, and Message Name parameters under one unique process request. A single run control ID can encompass multiple request IDs.

Parameters

Select the name of the message to publish.

The PeopleSoft system adds a run control for the currently effective FullSync message that is chunked by the setID CUSTOMER_FULLSYNC_EFF_SETID.

Note: If you insert a new row, the same run control component can publish more than one message, so you can produce both the full message and the current effective-dated full message from the same PeopleSoft Process Scheduler run.

Note: You must set up the run control parameters to start the Full Data Publish program.

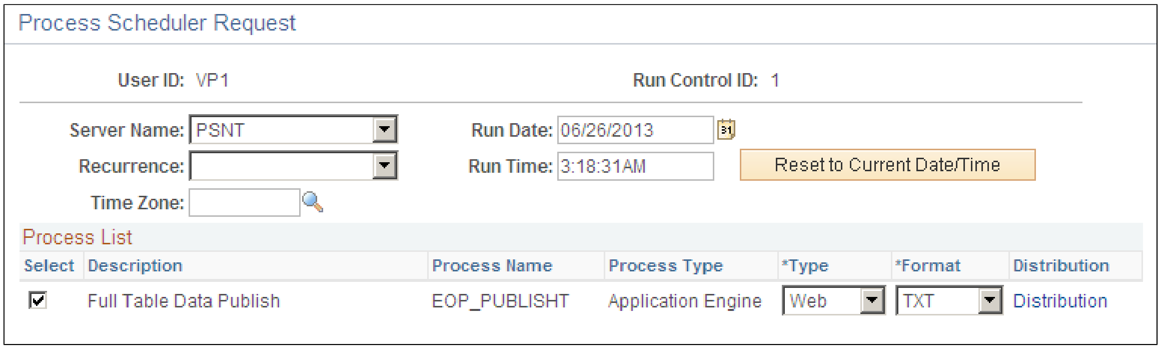
Performing Full Table Replication

Access the Full Data Publish page (Enterprise Components, Integration Definitions, Initiate Processes, Full Data Publish).

Click the Run button on the Full Data Publish (EO_FULLDATAPUB) page to access and select the Full Table Data Publish check box and perform a full table replication.

Image: Process Scheduler Request page showing the Full Table Data Publish option

This example illustrates the fields and controls on the Process Scheduler Request page showing the Full Table Data Publish option. You can find definitions for the fields and controls later on this page.



Process Scheduler Request

User ID: VP1

Run Control ID: 1

Server Name: PSNT

Run Date: 06/26/2013

Recurrence:

Run Time: 3:18:31AM

Time Zone:

Process List

Select	Description	Process Name	Process Type	*Type	*Format	Distribution
☑	Full Table Data Publish	EOP_PUBLISHT	Application Engine	Web	TXT	Distribution

Defining Routing

To define the message routing, you must create an OnRouteSend service operation handler and define the handler in the service operation.

Note: Perform this step only if you're chunking a message.

Example

You want to route the customer message to the nodes that are defined in the SetID/Nodes page within the Publish Setup component. Add the following PeopleCode to the OnRouteSend handler for the service operation:

```
Declare Function GETSETIDNODES PeopleCode FUNCLIB_EOEIP.PUBLISH_ROUTE_PC
     Field Formula;
Local Message &MSG;
/* Call Function that looks at Setid of first transaction in the message
 and returns a list of subscribing nodes to route the message */
&MSG = GetMessage();
GETSETIDNODES(&MSG, %Date);
```

Publishing Incremental Messages of Current Effective Data

This section discusses how to:

- Create service operations for publishing incremental messages of current effective data.
- Create subscription processes that open the generic effective-dated delay function.

For incremental messages, use subscription PeopleCode to copy current effective rows to a current incremental message for immediate publication, to strip out historic data, and to store future effective rows in a delay table. A regularly scheduled Application Engine program uses the delay table data as a trigger to publish future data when that data becomes effective.

Creating Service Operations for Publishing Incremental Messages of Current Effective Data

You must create a service operation and associate an incremental message definition with it.

Note: The name of the service operation must end with the suffix *_EFF* to be effective dated.

The incremental message definition must contain the necessary records in the publishing system. Specify the message version first, otherwise you can't add the tables that compose the message. When the system requests a message channel, enter the queue definition that you previously selected.

Creating Subscription Processes That Open the Generic Effective-Dated Delay Function

The PeopleSoft system includes a function called PROCESS_EFFDT_MSG. This function reads through an incremental message and processes the past, current, future, or non-effective-dated information. PROCESS_EFFDT_MSG resides within the record FUNCLIB_EOEIP, in the EFFDT_MSG_PC field.

To create a subscription process that opens the generic effective-dated Delay function:

1. Open the Process_Effdt_Msg generic function stored in record FUNCLIB_EOEIP.
2. Pass it the name of the current effective-dated incremental message.
3. Indicate whether only rows with *Active* effective status should be selected as the current effective-dated rows.

If the &ACTIVE_EFFSTATUS parameter passed in is set to *False,* the current effective-dated row (whether active or inactive) is selected.

4. Pass the parameter set to *True* if only active effective-dated rows should be sent to the other system.

Note: The standard setting for the &ACTIVE_EFFSTATUS parameter is *False.*

Publishing Effective-Dated Rows from the Delay Table

This section provides an overview of effective-dated row publishing from the delay table and discusses how to run the Effective Date Publish utility.

Page Used to Publish Effective-Dated Rows from the Delay Table

Page Name	***Definition Name***	***Navigation***	***Usage***
Effective Date Pub	EO_EFFDATAPUB	Enterprise Components, Integration Definitions, Initiate Processes, Effective Date Publish	Run the Effective Date Publish utility.

Understanding Effective-Dated Row Publishing

Publishing effective-dated rows from a delay table requires:

- A future-dated entry in the delay table.
- The process request page.
- An Application Engine utility program that publishes future-dated information when it becomes current.

The Application Engine Effective Date Publish utility publishes effective-dated rows from a delay table by:

1. Retrieving from the delay table any entries that are effective within a date range.
2. Using the key strings from the delay table and record information for the current message to read the original application tables and retrieve the most current effective rows.
3. Publishing those rows to the current incremental message.

   The third-party application subscribes to the current incremental messages.

The PeopleSoft system allows for an end date range on the Effective Date Publish utility if the utility was not run for one day. The end date range enables the program to run on the next date.

You can run the Effective Date Publish utility multiple times during the day, and it deletes the information from the delay queue when the future effective date data becomes current and the message for that data has been published. The Effective Date Publish utility retrieves only the latest delay table information since the prior run.

If the Effective Date Publish utility is invoked after not running for a period of time, it retrieves only the current row and publishes that as the active record. The presumption is that the subscribers want only the most current database information that is published.

Effective Date Pub Page

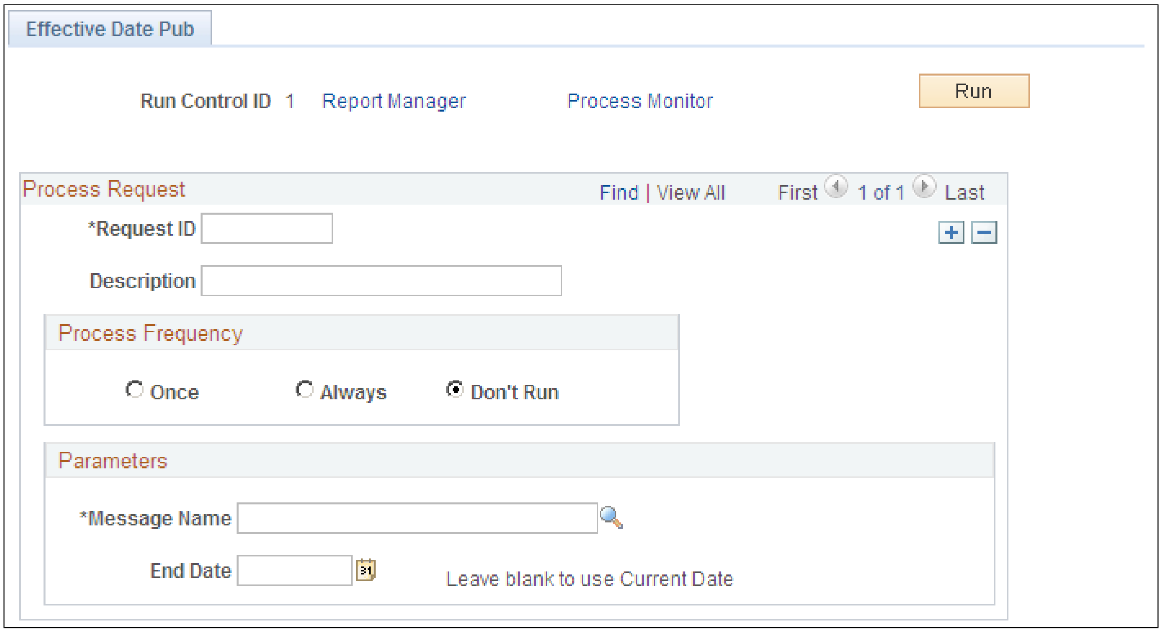
Use the Effective Date Pub page (EO_EFFDATAPUB) to run the Effective Date Publish utility.

Navigation

Enterprise Components, Integration Definitions, Initiate Processes, Effective Date Publish

Image: Effective Date Pub page

This example illustrates the fields and controls on the Effective Date Pub page. You can find definitions for the fields and controls later on this page.



Run Control ID 1

Process Request 1 of 1

*Request ID

Description

Process Frequency

○ Once ○ Always ● Don't Run

Parameters

*Message Name

End Date Leave blank to use Current Date

Message Name Select the current incremental message to publish.

End Date Select the latest effective date to process from the delay table.

Run Click to run this request.

The Application Engine program uses the trigger records in the delay table and the end date parameter from the run control component to publish a current effective incremental message that contains all future-dated rows that are effective. This ensures that third-party systems that cannot manage future-dated records always receive currently active data on that data's effective date, even if that information was previously updated on the PeopleSoft system.

Publishing Effective-Dated Rows and Prior-Dated Rows from the Delay Table

This section provides an overview of publishing effective-dated rows and prior-dated rows from the Delay table, and discusses how to run the Effective Date and Prior Publish utility.

Page Used to Publish Effective-Dated Rows and Prior-Dated Rows from the Delay Table

Page Name	***Definition Name***	***Navigation***	***Usage***
Effective Date Prior Publish	EO_EFFPRIORPUB	Enterprise Components, Integration Definitions, Initiate Processes, Effective Date Prior Publish	Run the Effective Date and Prior Publish utility.

Understanding Publishing Effective-Dated Rows and Prior-Dated Rows from the Delay Table

Publishing effective-dated rows and prior-dated rows from the delay table is similar to publishing effective-dated rows. The difference is that in addition to publishing messages with a specified effective date or date range, the system also publishes all prior-dated rows with non-key changes.

Related Links

Understanding Effective-Dated Row Publishing

Effective Date Prior Publish Page

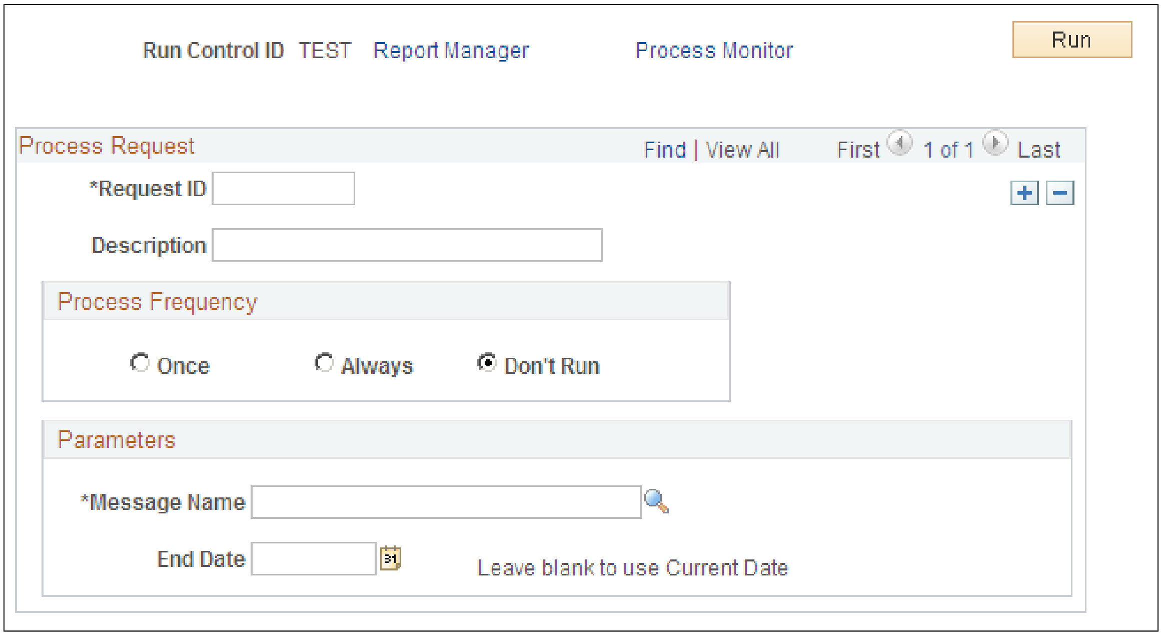
Use the Effective Date Prior Publish page (EO_EFFPRIORPUB) to run the Effective Date and Prior Publish utility.

Navigation

Enterprise Components, Integration Definitions, Initiate Processes, Effective Date Prior Publish

Image: Effective Date Prior Publish page

This example illustrates the fields and controls on the Effective Date Prior Publish page. You can find definitions for the fields and controls later on this page.



Run Control ID TEST

Process Request

*Request ID

Description

Process Frequency

○ Once ○ Always ● Don't Run

Parameters

*Message Name

End Date Leave blank to use Current Date

Message Name Select the current incremental message to publish.

End Date Select the highest effective date to process from the delay table.

Run Click to run this request.

Chapter 8

Using the Flat File Utility

Understanding the Flat File Utility

When external systems send flat files to you for inbound transactions, you must develop complementary processes to translate incoming files into messages or translate outbound messages into files.

This is the flow for inbound file processing when you use the Flat File utility:

1. The utility receives a flat file in the form of a file layout object from an external system.

   The flat file consists of one of the following items:

   - The relevant data.
   - An index file that contains pointers to the data.

     Each index file lists the names of a set of data files to be processed. These files contain the application data, which is in one of the following formats: fixed record, Comma Separated Values (CSV), or XML.

2. The utility reads the file that is submitted for processing:

   - If the file is an index file, the Flat File utility loads the list of data files that are associated with each index file to be processed into a parameter table.
   - If it is a single data file, the utility inserts the single data file into a parameter table.

     Note: If additional fields in the file layout are not in the message definition, the additional fields are ignored during the copying of the flat file data to the message and are not included in the message.

3. The utility loops through the list of data files to be processed and reads each data file.
4. The utility copies the row sets of the data files into the message.
5. The utility publishes the message.
6. The subscribing systems receive the message and initiate normal inbound data processing.

Processing Inbound Flat Files

You use the file layout definition to read and write from flat files.

To process inbound flat files:

1. Determine the necessary format of the inbound data.

   If an industry standard exists, use it for your file definition.

   If no industry standard exists, create a file layout object that mirrors your message object.

2. Identify the inbound process and its standard message.

3. Analyze the vendor's file structure and compare it to the standard message.

   Answer these questions:

   - Can you use an existing message, or do you need to create a new one?
   - Can the customer conform to an existing integration point, or do you need to create one (along with corresponding subscription PeopleCode)?

4. Create the message definition.

5. Create a file layout definition with the same structure as the message definition to support the vendor file format.

   The hierarchical structure of the data in the File Layout Definition must match that of the message definition. For example, suppose a message has three levels: level zero, containing record A, level one, containing records B and C, and level two, containing record D.

   All file layouts that are associated with this message must also have record A in level zero, record B and C in level one, and record D in level two.

   Note: The file layout does not need to contain the exact same fields as the message definition.

   For every record in your file layout, add a new file field, AUDIT_ACTN, as the first field in the record (except when the field already exists in the application table).

   You can associate more than one file layout with a single message. For example, vendor A may have a different number of fields than vendor B, so you may have two file layouts: one for A and one for B.

   Specify the file ID uniquely to include a row in a file, which is necessary in mapping the data to its proper record. Include start and end points when dealing with more than one record in a file layout.

   Note: Each record in the file layout has a file record ID attribute. Do not confuse this with the file layout ID. The file layout ID determines whether a new layout is encountered for multiple file layout processing.

   When you subscribe to the message and normal inbound data processing begins, you can invoke the SetDefault PeopleCode function to set the default values for fields that were not present in the input file.

6. Update or create the inbound file rule pages.

7. Create subscription PeopleCode in PeopleSoft Application Designer to process the message.

Have the standard inbound process subscribe to and process the message normally. The standard message definition should have a subscription process that initiates the normal inbound processing for the object to which you hook your application logic to process the file data.

8. Test the inbound flat file processing.

Note: You can process multiple inbound flat files at one time. Specifying an inbound index file as part of the Flat File utility parameters causes the system to read all input files within the index file and to use the associated file layout object and message to convert the data. Similarly, specify a wildcard in the filename in the inbound file rule component, but make sure that all files that meet the wildcard criteria correspond to the file layout and message mapping that are defined.

Initiating File Processing

This section discusses how to:

- Set up inbound flat file processing.
- Initiate inbound flat file processing.

Pages Used to Initiate File Processing

Page Name	***Definition Name***	***Navigation***	***Usage***
File Inbound	EO_FILE_INBOUND	Enterprise Components, Integration Definitions, Inbound File Rule	Set up inbound flat file processing.
Inbound File	EO_FILETOMSG	Enterprise Components, Integration Definitions, Initiate Processes, Inbound File Publish	Initiate inbound flat file processing. This file-to-message processing function reads the file rowset and publishes it as a message.

File Inbound Page

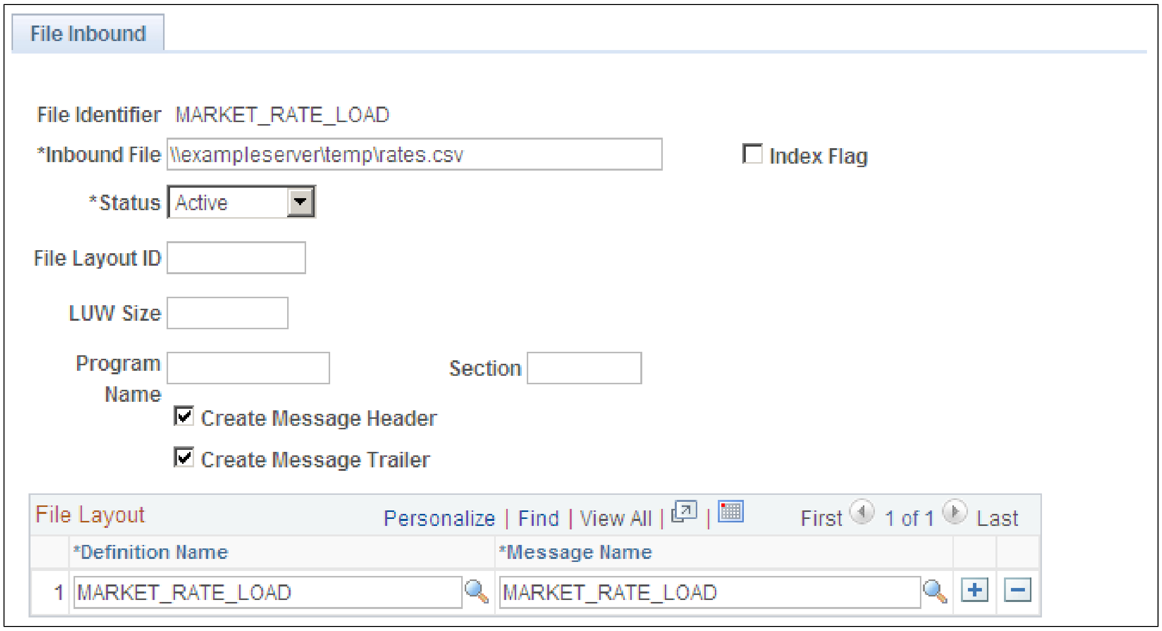
Use the File Inbound page (EO_FILE_INBOUND) to set up inbound flat file processing.

Navigation

Enterprise Components, Integration Definitions, Inbound File Rule

Image: File Inbound page

This example illustrates the fields and controls on the File Inbound page. You can find definitions for the fields and controls later on this page.



File Identifier MARKET_RATE_LOAD

*Inbound File \\exampleserver\temp\rates.csv

□ Index Flag

*Status Active

File Layout ID

LUW Size

Program Name

Section

☑ Create Message Header

☑ Create Message Trailer

	*Definition Name	*Message Name
1	MARKET_RATE_LOAD	MARKET_RATE_LOAD

Referencing Application Engine Program

The file inbound setup allows you to reference an AE program and section that gets called as soon as the data processing of the specified inbound file is completed. This functionality comes in handy if you wish to invoke customized actions after the flat file utility finishes processing data--simply build the custom AE program and specify it on this page for the corresponding file rule.

Below is a sample code for an AE program that removes entries of data files that have been processed from index files. Steps of this program are executed only if the input file is flagged as an Index file.

```
/* Open the index file in Read-only mode */
/* Initialize a string array and save all unprocessed file names from the Index fil⇒
e */
/* Open the Index in Write mode, thereby clearing contents from previous iteration,⇒
 and write the contents of the array to the file */
/* When the last data file has been processed, write 'All Done' to the Index file *⇒
/

Local string &Index_FileName, &LineString;
Local File &Index_File;
Local array of string &Inbound_Files;
Local number &I;
Local boolean &Found;
If EO_FILEPUB_AET.INDEX_FILE_FLG = "Y" Then
   /* Get index filename from the Inbound File Publish Rule */
   SQLExec("Select FILE_INBOUND from PS_EO_FLOINDEFN WHERE FILE_ID = :1", EO_   FIL⇒
EPUB_AET.FILE_ID, &Index_FileName);

   /* Open Index file as Read-only */
   &Index_File = GetFile(&Index_FileName, "R", %FilePath_Absolute);
```

```
    /* Initialise string array */
    &Inbound_Files = CreateArrayRept("", 0);

    /* Save unprocessed filenames to array */
    While &Index_File.ReadLine(&LineString);
       If &LineString <> EO_FILEPUB_AET.FILE_INBOUND Then
          &Inbound_Files.Push(&LineString);
       End-If;
    End-While;
    &Index_File.Close();

    /*  Open index file in Write mode */
    &Index_File = GetFile(&Index_FileName, "W", %FilePath_Absolute);

    /*  Write unprocessed filenames to index file  */
    &I = 0;
    While &Inbound_Files.Next(&I)
       &Found = True;
       &Index_File.WriteLine(&Inbound_Files [&I]);
    End-While;

    /* If all files processed, print Done */
    If Not &Found Then
       &Index_File.WriteLine("All Done!");
    End-If;

    &Index_File.Close();

End-If;
```

File Identifier

Displays the inbound file that you are associating with the rule.

Inbound File

Enter the index file name or the data file name. Specify the full path information. The PeopleCode program uses the *%filepath_ absolute* variable when opening the file.

Index Flag

Select to distinguish between the index and the data file.

Status

Select whether this inbound file rule is *Active* or *Inactive.* The default value is *Inactive.*

File Layout ID

Select a layout to associate with the file.

LUWSize (logical unit of work size)

To limit the message size, enter the number of level zero rows that are in each message. The output message is normally determined by the MaxMessageSize system parameter.

Program and Section

Enter the name of a PeopleSoft Application Engine (AE) program and section to invoke when the utility finishes processing data.

Create Message Header

Select to create a header message. Use the header message as a trigger in the subscription process to initialize tables before they receive the data messages. This option is selected by default.

Create Message Trailer

Select to create a trailer message. Use the trailer message as a trigger in the subscription process to indicate that all the data messages have been received. This option is selected by default.

File Layout

Definition Name and Message Name

If the File Layout ID field is blank, this field should contain only one entry.

If the File Layout ID field is not blank, this scroll area must contain an entry for each file layout definition name that is specified in the inbound file.

Note: Use the wildcards * and ? for the file name but not for the directory path. The file layout and message mapping must be valid for all files that meet the wildcard criteria.

Inbound File Page

Use the Inbound File page (EO_FILETOMSG) to initiate inbound flat file processing.

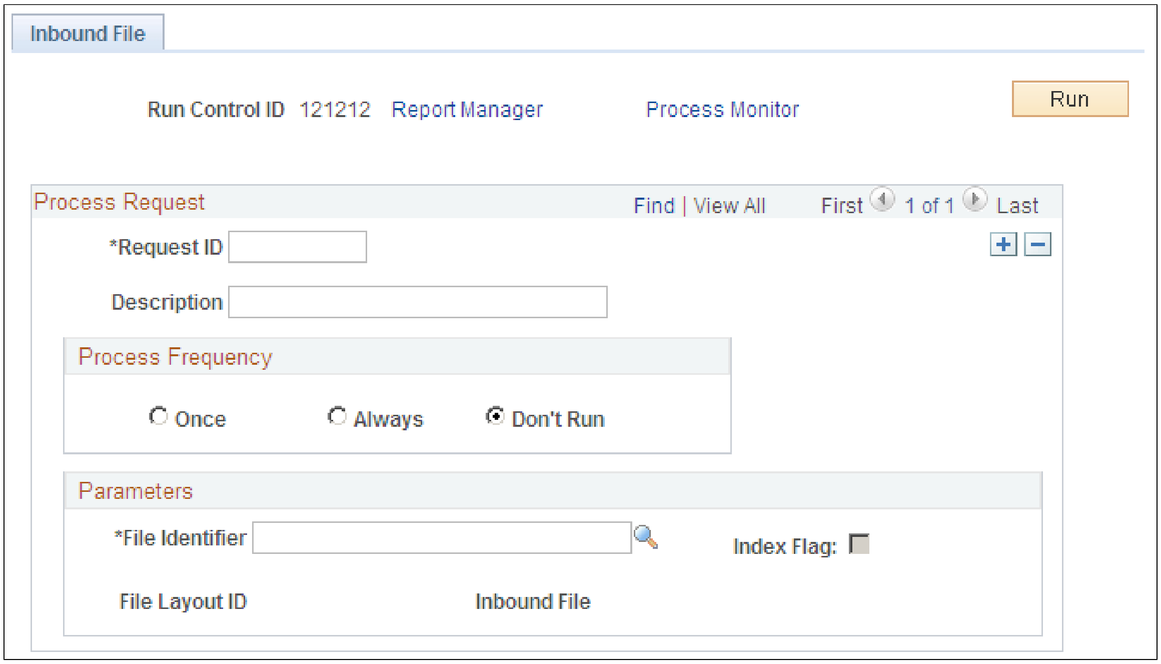
This file-to-message processing function reads the file rowset and publishes it as a message.

Navigation

Enterprise Components, Integration Definitions, Initiate Processes, Inbound File Publish

Image: Inbound File page

This example illustrates the fields and controls on the Inbound File page. You can find definitions for the fields and controls later on this page.



Run Control ID 121212

Process Request

*Request ID

Description

Process Frequency

○ Once ○ Always ● Don't Run

Parameters

*File Identifier

Index Flag: □

File Layout ID

Inbound File

Use this page to initiate inbound flat file processing. This file-to-message processing function reads the file rowset and publishes it as a message.

Parameters

File Identifier

Select or enter the name of the file identifier that you set up in the File Inbound page. The file identifier is tied to the publish rules.

Run

Click to run this request.

Publishing a New Message

The Inbound File page runs an Application Engine process that initiates the file-to-message processing. The file-to-message processing function reads the file rowset and publishes it as a message.

If an index file exists when the inbound conversion process runs, the Application Engine program loads the list of files to be converted into a parameter table and completes a commit. The Application Engine program uses the list of files within the parameter table to restart the processing if a particular flat file fails. If a single data file is provided, then the rowset processing immediately begins.

The file publish process goes through each of the rowsets of the file layout and copies them into the message row sets.

If the audit action (AUDIT_ACTN) exists in the file, it is copied to the PSCAMA record. If the audit action does not exist in the file, the publishing process uses the default value that is specified in the file layout field property.

The Flat File utility publishes a new message when one of the following situations occurs:

- Maximum message size is exceeded.
- Logical unit of work publish size is reached.
- A new file layout is detected.
- End of file is reached.

The Application Engine program completes a commit every time a message is published from a file. After conversion, the flat file remains in the parameter table with a status of *Processed.*

Note: The file layout should exactly match the message layout (excluding the PSCAMA record) and should use the same character set as that used by the file: either American National Standards Institute or Unicode.

Testing Inbound Flat File Processing

To test inbound files:

1. Create a sample flat file, or ask the third-party vendor for a sample flat file.
2. Launch the Flat File utility.
    a. Through the browser, sign in to PeopleSoft Internet Architecture.

  b. Select Enterprise Components, Management, Inbound File Rule.

3. Run the Application Engine program to convert the sample flat file to a message by running Message Monitor.

   Use Message Monitor to ensure that the inbound file processing created a publish message that contains the sample flat file data.

   a. Verify that the standard inbound subscription process received the message and processed it into the application tables.

   b. Determine whether the values become the inherited values (if you used the inherited value feature in file layout).

   c. Validate that the production or staging tables loaded with the correct field values.

      For production tables, look in the PeopleSoft application pages.

      For staging tables, use either the PeopleSoft application pages or run a query by using PeopleSoft Query.

   d. Ensure that the date formats conform.

Chapter 9

Using the XML Schema Utility

Understanding the XML Schema Utility

PeopleSoft Open Integration Framework enables near real-time messaging and transactions by using a format that is based on XML to convey information between diverse applications in a standard way. To take advantage of this standardization, you must obtain clear XML definitions (schemas) for each application message, component interface, or business interlink.

The XML Schema utility provides the following features:

- Output options for the XML Schema utility, document type definition (DTD), or BizTalk definition for all application messages.
- The ability to create an XML definition for a single object, for all of the objects, or for all of the objects by a specific owner.
- A single flat file for each XML definition that is written to your system's %TEMP directory (when you use the Microsoft Windows client) or the server's common access file directory (when you use PeopleSoft Internet Architecture).
- An application foundation for future standards of XML definitions.

Generating the XML Schema

This section discusses how to generate the XML Schema.

Page Used to Generate the XML Schema

Page Name	***Definition Name***	***Navigation***	***Usage***
Generate XML Schema	EO_GEN_XML_DATA	Enterprise Components, Integration Definitions, Review XML Schema	Generate DTDs, XML schemas, and BizTalk definitions.

Generate XML Schema Page

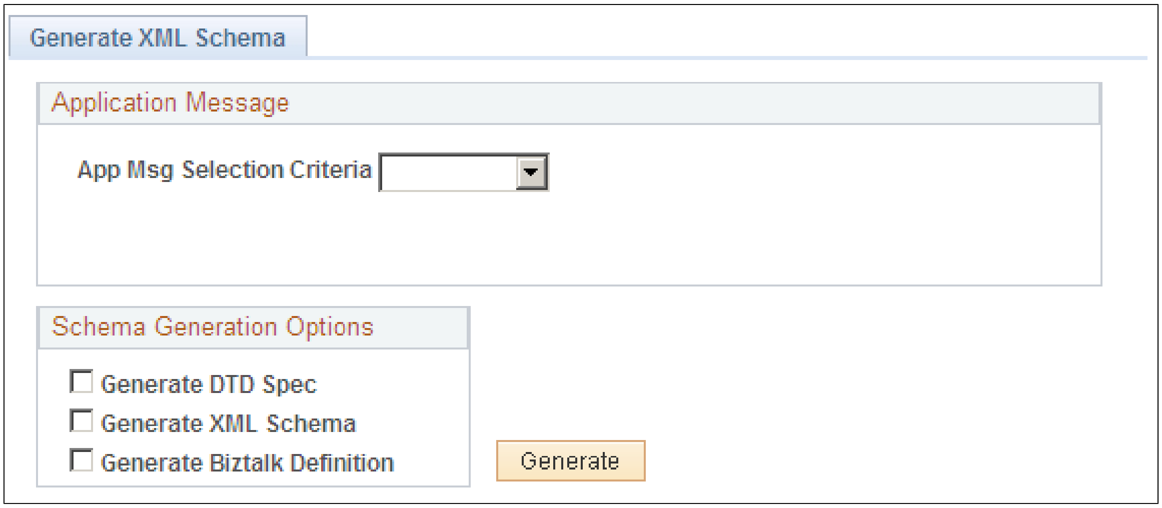
Use the Generate XML Schema page (EO_GEN_XML_DATA) to generate DTDs, XML schemas, and BizTalk definitions.

Navigation

Enterprise Components, Integration Definitions, Review XML Schema

Image: Generate XML Schema page

This example illustrates the fields and controls on the Generate XML Schema page. You can find definitions for the fields and controls later on this page.



App Msg Selection Criteria (application message selection criteria) Select application selection message criteria. Values are *All Msg* (all messages), *Channel, Owner ID* and *Single Msg* (single message). Depending on the selection criteria, you can enter the selection value (if already known) or search for the value.

Generate DTD Spec Select for DTD spec output format.

Generate XML Schema Select for XML schema output format.

Generate Biztalk Definition Select for BizTalk definition output format.

Generate Click to generate the selected output formats.

The utility queries the relevant PeopleTools tables to generate the selected types of XML schemas and writes the results to the server's file directory or your system's Temp directory, depending on the client that you use.

To produce an XML schema, DTD, or BizTalk definition:

1. Define the selection criteria for application messages, component interfaces, and business interlinks.
2. Select XML schemas, DTDs, or BizTalk definitions for application message, component interface, and business interlink objects.

   PeopleCode sends a query to the PeopleTools tables to create the selected types of XML definitions.

   The XML Schema utility then writes the definitions to the file directory as specified by the PS_SERVDIR environment variable when you are using PeopleSoft Internet Architecture or the %TEMP directory of your system when you are using the Microsoft Windows client.

Interpreting Sample Output

The following code shows three samples of output for the same application message (in this case, MARKET_RATE_TYPE_FULLSYNC) in BizTalk, DTD, and XML schema formats.

Example: BizTalk

The following code shows MARKET_RATE_TYPE_FULLSYNC in BizTalk format:

```
<BizTalk xmlns="urn:schemas-biztalk-org:BizTalk/biztalk-0.81.xml">
<Body>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:element name="MARKET_RATE_TYPE_FULLSYNC"
  type="MARKET_RATE_TYPE_FULLSYNCType"/>

<xsd:complexType name="MARKET_RATE_TYPE_FULLSYNCType">
<xsd:sequence>
   <xsd:element name="FieldTypes" type="FieldTypesType"/>
   <xsd:element name="MsgData" type="MsgDataType"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesType">
<xsd:sequence>
   <xsd:element name="RT_TYPE_TBL" type="FieldTypesRT_TYPE_TBLType"/>
   <xsd:element name="PSCAMA" type="PSCAMA"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE" type="FieldTypesFieldType"/>
   <xsd:element name="DESCR" type="FieldTypesFieldType"/>
   <xsd:element name="DESCRSHORT" type="FieldTypesFieldType"/>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="PSCAMA">
   <xsd:sequence>
   <xsd:element name="LANGUAGE_CD" type="LANGUAGE_CDType" minOccurs="0"
     maxOccurs "1"/>
   <xsd:element name="AUDIT_ACTN" type="AUDIT_ACTNType"/>
   <xsd:element name="BASE_LANGUAGE_CD" type="BASE_LANGUAGE_CDType"
     minOccurs="0"maxOccurs="1"/>
   <xsd:element name="MSG_SEQ_FLG" type="MSG_SEQ_FLGType" minOccurs="0"
     maxOccurs= "1"/>
   <xsd:element name="PROCESS_INSTANCE" type="PROCESS_INSTANCEType"
     minOccurs="0" maxOccurs="1"/>
   <xsd:element name="PUBLISH_RULE_ID" type="PUBLISH_RULE_IDType"
     minOccurs="0" maxOccurs="1"/>
   <xsd:element name="MSGNODENAME" type="MSGNODENAMEType" minOccurs="0"
     maxOccurs= "1"/>
   </xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="LANGUAGE_CDType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="AUDIT_ACTNType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="BASE_LANGUAGE_CDType"  >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>
```

```
<xsd:complexType name="MSG_SEQ_FLGType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PROCESS_INSTANCEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PUBLISH_RULE_IDType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSGNODENAMEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>


<xsd:complexType name="MsgDataType">
      <xsd:sequence>
   <xsd:element name="Transaction">
      <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="RT_TYPE_TBL" type="MsgDataRT_TYPE_TBLType"/>
         <xsd:element name="PSCAMA" type="PSCAMA"/>
      </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
      </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MsgDataRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE">
<xsd:simpleType>
<xsd:restriction base="xsd:string" >
   <xsd:pattern value="[A-Z]{1-5}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCR">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-30}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCRSHORT">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-10}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="FieldTypesFieldType">
   <xsd:attribute name="type" type="fieldtypes"/>
</xsd:complexType>

<xsd:simpleType name="fieldtypes">
<xsd:restriction base="xsd:string">
   <xsd:enumeration value="CHAR"/>
   <xsd:enumeration value="NUMBER"/>
   <xsd:enumeration value="DATE"/>
   <xsd:enumeration value="DATETIME"/>
   <xsd:enumeration value="TIME"/>
</xsd:restriction>
</xsd:simpleType>
```

```
</xsd:schema>
</Body>
</BizTalk>
```

Example: DTD

The following code shows MARKET_RATE_TYPE_FULLSYNC in DTD format:

```
<!ELEMENT MARKET_RATE_TYPE_FULLSYNC (FieldTypes, MsgData)>

   <!ENTITY % recordtypes "class (R | SR) #REQUIRED" >
   <!ENTITY % fieldtypes "type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED" >

<!ELEMENT FieldTypes (RT_TYPE_TBL, PSCAMA)>

<!ELEMENT PSCAMA (LANGUAGE_CD?, AUDIT_ACTN, BASE_LANGUAGE_CD?, MSG_SEQ_FLG?,

 PROCESS_INSTANCE?, PUBLISH_RULE_ID?, MSGNODENAME?)>
   <!ATTLIST PSCAMA class (R | SR) #REQUIRED>
   <!ELEMENT LANGUAGE_CD (#PCDATA)>
      <!ATTLIST LANGUAGE_CD type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT AUDIT_ACTN (#PCDATA)>
      <!ATTLIST AUDIT_ACTN type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT BASE_LANGUAGE_CD (#PCDATA)>
      <!ATTLIST BASE_LANGUAGE_CD type (CHAR | NUMBER | DATE | TIME |
        DATETIME) #IMPLIED>
   <!ELEMENT MSG_SEQ_FLG (#PCDATA)>
      <!ATTLIST MSG_SEQ_FLG type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT PROCESS_INSTANCE (#PCDATA)>
      <!ATTLIST PROCESS_INSTANCE type (CHAR | NUMBER | DATE | TIME |
        DATETIME) #IMPLIED>
   <!ELEMENT PUBLISH_RULE_ID (#PCDATA)>
      <!ATTLIST PUBLISH_RULE_ID type (CHAR | NUMBER | DATE | TIME |
        DATETIME) #IMPLIED>
   <!ELEMENT MSGNODENAME (#PCDATA)>
      <!ATTLIST MSGNODENAME type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
<!ELEMENT RT_TYPE_TBL (RT_TYPE, DESCR, DESCRSHORT)>
   <!ATTLIST RT_TYPE_TBL class (R | SR) #REQUIRED>

<!ELEMENT RT_TYPE (#PCDATA)>
      <!ATTLIST RT_TYPE type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT DESCR (#PCDATA)>
      <!ATTLIST DESCR type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>
   <!ELEMENT DESCRSHORT (#PCDATA)>
      <!ATTLIST DESCRSHORT type (CHAR | NUMBER | DATE | TIME | DATETIME) #IMPLIED>

<!ELEMENT MsgData (Transaction)>
<!ELEMENT Transaction (RT_TYPE_TBL, PSCAMA)>
```

Example: XML Schema

The following code shows MARKET_RATE_TYPE_FULLSYNC in XML schema format:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">

<xsd:element name="MARKET_RATE_TYPE_FULLSYNC"
  type="MARKET_RATE_TYPE_FULLSYNCType"/>

<xsd:complexType name="MARKET_RATE_TYPE_FULLSYNCType">
<xsd:sequence>
   <xsd:element name="FieldTypes" type="FieldTypesType"/>
   <xsd:element name="MsgData" type="MsgDataType"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesType">
<xsd:sequence>
   <xsd:element name="RT_TYPE_TBL" type="FieldTypesRT_TYPE_TBLType"/>
   <xsd:element name="PSCAMA" type="PSCAMA"/>
```

```
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldTypesRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE" type="FieldTypesFieldType"/>
   <xsd:element name="DESCR" type="FieldTypesFieldType"/>
   <xsd:element name="DESCRSHORT" type="FieldTypesFieldType"/>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="PSCAMA">
   <xsd:sequence>
   <xsd:element name="LANGUAGE_CD" type="LANGUAGE_CDType" minOccurs="0"
     maxOccurs= "1"/>
   <xsd:element name="AUDIT_ACTN" type="AUDIT_ACTNType"/>
   <xsd:element name="BASE_LANGUAGE_CD" type="BASE_LANGUAGE_CDType"
     minOccurs="0" maxOccurs="1"/>
   <xsd:element name="MSG_SEQ_FLG" type="MSG_SEQ_FLGType" minOccurs="0"
     maxOccurs= "1"/>
   <xsd:element name="PROCESS_INSTANCE" type="PROCESS_INSTANCEType"
     minOccurs="0" maxOccurs="1"/>
   <xsd:element name="PUBLISH_RULE_ID" type="PUBLISH_RULE_IDType"
     minOccurs="0" maxOccurs="1"/>
   <xsd:element name="MSGNODENAME" type="MSGNODENAMEType" minOccurs="0"
     maxOccurs= "1"/>
   </xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="LANGUAGE_CDType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="AUDIT_ACTNType" >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="BASE_LANGUAGE_CDType"  >
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSG_SEQ_FLGType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PROCESS_INSTANCEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="PUBLISH_RULE_IDType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>

<xsd:complexType name="MSGNODENAMEType">
   <xsd:attribute name="type" type="fieldtypes" use="optional"/>
</xsd:complexType>


<xsd:complexType name="MsgDataType">
      <xsd:sequence>
   <xsd:element name="Transaction">
      <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="RT_TYPE_TBL" type="MsgDataRT_TYPE_TBLType"/>
         <xsd:element name="PSCAMA" type="PSCAMA"/>
      </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
      </xsd:sequence>
```

```
</xsd:complexType>

<xsd:complexType name="MsgDataRT_TYPE_TBLType">
<xsd:sequence>
   <xsd:element name="RT_TYPE">
<xsd:simpleType>
<xsd:restriction base="xsd:string" >
   <xsd:pattern value="[A-Z]{1-5}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCR">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-30}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
   <xsd:element name="DESCRSHORT">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
   <xsd:pattern value=".{1-10}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
   <xsd:attribute name="class" type="xsd:string" use="required" value="R"/>
</xsd:complexType>

<xsd:complexType name="FieldTypesFieldType">
   <xsd:attribute name="type" type="fieldtypes"/>
</xsd:complexType>

<xsd:simpleType name="fieldtypes">
<xsd:restriction base="xsd:string">
   <xsd:enumeration value="CHAR"/>
   <xsd:enumeration value="NUMBER"/>
   <xsd:enumeration value="DATE"/>
   <xsd:enumeration value="DATETIME"/>
   <xsd:enumeration value="TIME"/>
</xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

Chapter 10

Using Desktop Integration

Understanding Desktop Integration

Desktop Integration provides seamless integration between Microsoft Office and your PeopleSoft applications, enabling your organization to schedule and track appointments, meetings, tasks, and corresponding documentation. This is accomplished through a framework to the Microsoft Office suite that enables PeopleSoft applications to extend functionality into the office suite through layers. A two-way communication is established between the PeopleSoft application and Microsoft Outlook and Word, with Outlook receiving the information from and sending updates to the PeopleSoft application.

The framework consists of the following components:

Microsoft Outlook and Word Add-In

Add-ins are supplemental programs that extend the functionality of Office applications, adding custom commands and custom features that plug into the application through well-known entry points. The Microsoft Outlook and Word Add-ins contain all of the pieces necessary for sending information to and receiving information from the PeopleSoft applications. Microsoft Office Add-ins invoke functions in this layer for user authentication, launching web services to receive and send data from the PeopleSoft application. On the PeopleSoft side, services and application classes exist that accept the requests from Microsoft Outlook and Word and perform the necessary actions and data retrieval. The Microsoft Outlook and Word Add-ins use a Secure Socket Layer (SSL) to connect to the PeopleSoft applications for secure and synchronous messaging.

See Understanding the PeopleSoft and Outlook Integration.

Integration Broker/Web Services Layer

Oracle uses the Integration Broker to connect Microsoft Outlook and Word to the PeopleSoft applications. Oracle delivers predefined interfaces and security classes. Desktop Integration does *not* require a dedicated server nor any additional hardware. To connect Microsoft Outlook and Word to PeopleSoft applications and ensure security, SSL is required.

Desktop Integration Layer

The Desktop Integration application classes are responsible for receiving the messages sent through Integration Broker from the Microsoft Office applications. The Desktop Integration layer creates, updates, and moves the data in and out of the PeopleSoft system and is responsible for responding to all events. As events are processed, control is optionally passed on to the PeopleSoft Application Layer for handling.

PeopleSoft Application Layer—for the PeopleSoft Applications to Use

Any application-specific functions are contained within the PeopleSoft Application Layer. When you are creating new integrations or customizing existing ones, all application-specific code is contained within the application layer.

Prerequisites

These are the prerequisites for using Desktop Integration:

- Microsoft Outlook 2007 or higher.

  You must license and install Microsoft Outlook 2007 or higher. Oracle does not provide this software, you must purchase and license it separately.

- Microsoft Word 2007 or higher.

  You must license and install Microsoft Word 2007 or higher. Oracle does not provide this software, you must purchase and license it separately.

- Integration Broker.

  You do not need a dedicated server. Oracle delivers the interface predefined. Your Integration Broker must be configured and started.

- Secure Socket Layer (SSL).

  This is used to connect to Microsoft Outlook. This is in the form of an https URL. Desktop Integration requires a secure connection for all communication.

- Unique System IDs.

  System IDs must be unique across your organization to allow use against multiple databases.

Setting Up Desktop Integration

This section discusses how to set up Desktop Integration. Specifically, it discusses how to:

- Set up the Integration Broker.
- Set up pages and web service security.
- Set up the system.

Pages Used to Set Up Desktop Integration

Page Name	***Definition Name***	***Navigation***	***Usage***
Service Operations - General page	IB_SERVICE	PeopleTools, Integration Broker, Integration Setup, Service Operations, General	Customize integration metadata to suit your particular business needs. However, note that integration metadata for this functionality (for example, service operations, routing definitions, service operations handlers, and so on) is delivered with your application; no additional configuration is required.

Page Name	***Definition Name***	***Navigation***	***Usage***
Web Service Access	WS_ACCESS_IB	Click the Service Operation Security link on the Service Operations - General page.	Define permissions and user access. Customize integration metadata to suit your particular business needs.
System Setup	EODI_SYSTEM_SETUP	Enterprise Components, Desktop Integration, System Setup	Configure Desktop Integration and the PeopleSoft system.

Service Operations - General Page

Use the Service Operations - General page (IB_SERVICE) to customize integration metadata to suit your particular business needs.

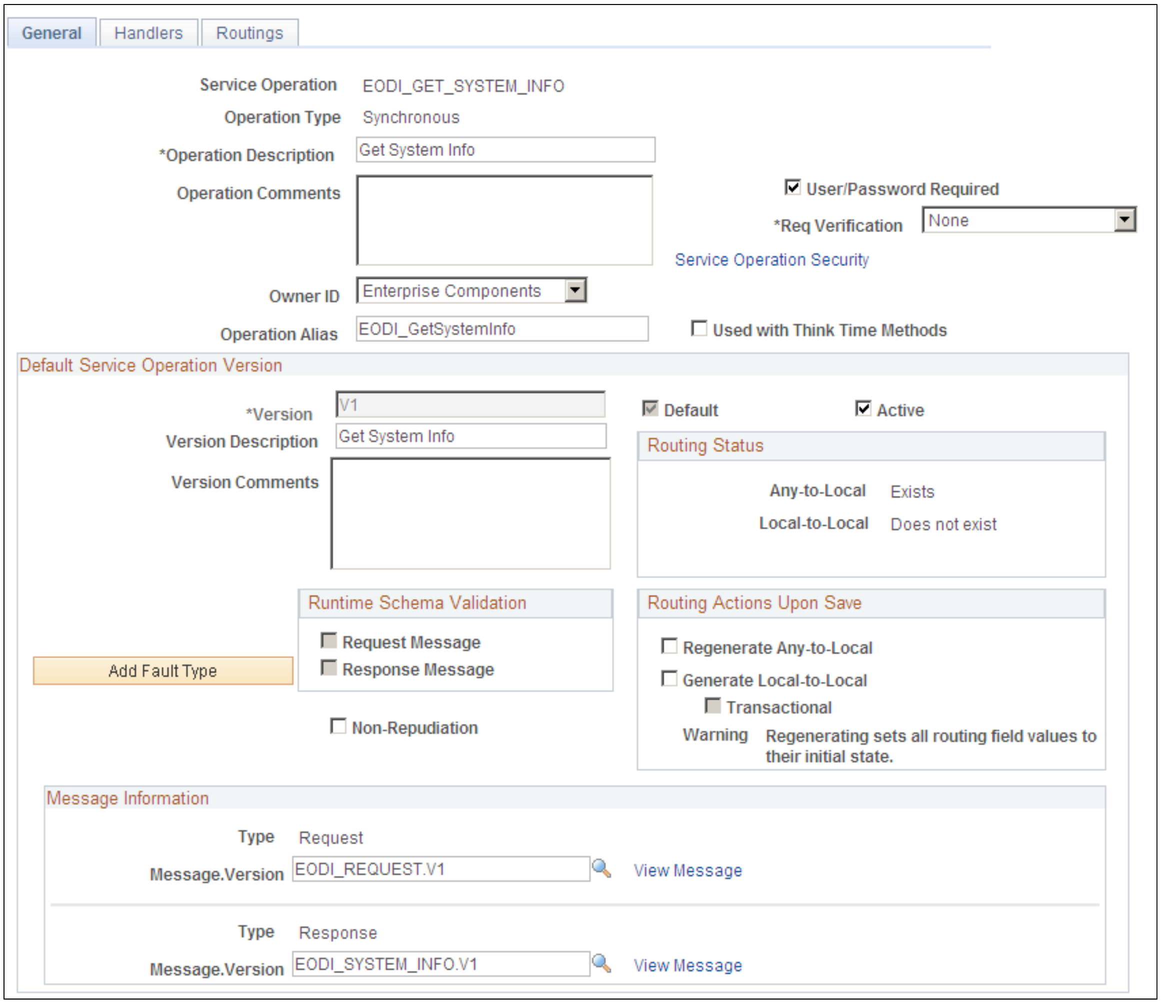
However, note that integration metadata for this functionality (for example, service operations, routing definitions, service operations handlers, and so on) is delivered with your application; no additional configuration is required.

Navigation

PeopleTools, Integration Broker, Integration Setup, Service Operations, General

Image: Service Operations - General page

This example illustrates the fields and controls on the Service Operations - General page. You can find definitions for the fields and controls later on this page.



Service Operation EODI_GET_SYSTEM_INFO

Operation Type Synchronous

*Operation Description Get System Info

Operation Comments

☑ User/Password Required

*Req Verification None

Owner ID Enterprise Components

Operation Alias EODI_GetSystemInfo

☐ Used with Think Time Methods

Default Service Operation Version

*Version V1

☑ Default

☑ Active

Version Description Get System Info

Version Comments

Routing Status

Any-to-Local Exists

Local-to-Local Does not exist

Runtime Schema Validation

☐ Request Message

☐ Response Message

☐ Non-Repudiation

Routing Actions Upon Save

☐ Regenerate Any-to-Local

☐ Generate Local-to-Local

☐ Transactional

Warning Regenerating sets all routing field values to their initial state.

Message Information

Type Request

Message.Version EODI_REQUEST.V1

Type Response

Message.Version EODI_SYSTEM_INFO.V1

To use Desktop Integration functionality, PeopleSoft Integration Broker must be set up and running.

Integration metadata for this functionality (for example, service operations, routing definitions, service operation handlers, and so on) is delivered with your application; no additional configuration is required. However, you can customize integration metadata to suit your particular business needs. All Desktop Integration service definitions have names that begin with *EODI* and are grouped by function. By default, the default local node receives all inbound integrations.

This product documentation contains all the information for configuring and setting up the Integration Broker.

See *PeopleTools: PeopleSoft Integration Broker Administration*

This product documentation describes how to create and modify integration metadata.

See *PeopleTools: PeopleSoft Integration Broker*

Web Service Access Page

Use the Web Service Access page (WS_ACCESS_IB) to define permissions and user access.

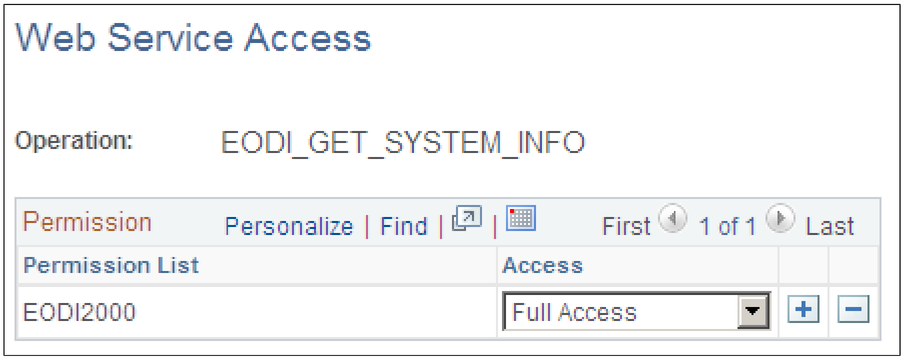
Customize integration metadata to suit your particular business needs.

Navigation

Click the Service Operation Security link on the Service Operations - General page.

Image: Web Service Access page

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.



Web Service Access

Operation: EODI_GET_SYSTEM_INFO

Permission List	Access
EODI2000	Full Access

Oracle delivers these permission lists:

EODI1000

Administrator level service operations for the queue. Select the access that you want for the user group:

- Full Access
- Queue
- App Class

EODI1000 is the permission list that contains all of the pages under the Desktop Integration menu. Administrators need access to this list.

EODI_Administrator has EODI1000 defined as a role.

EODI2000

Service level service operations for the application classes. Select the access that you want for the user group:

- Full Access
- Queue
- App Class

EODI2000 is the permission list that contains all of the web services that Desktop Integration uses. Any user of the Microsoft Outlook and Word Add-Ins needs access to this list.

The permission list contains a list of all Integration Broker services that the add-ins use to communicate with PeopleSoft applications.

EODI_RemoteUser has EODI2000 defined as a role.

System Setup Page

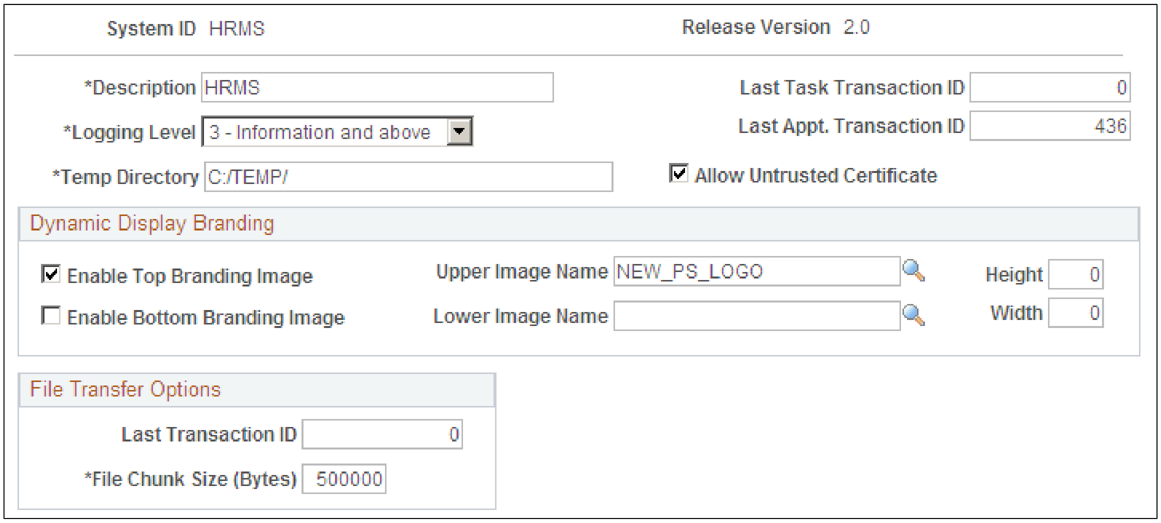
Use the System Setup page (EODI_SYSTEM_SETUP) to configure Desktop Integration and the PeopleSoft system.

Navigation

Enterprise Components, Desktop Integration, System Setup

Image: System Setup page

This example illustrates the fields and controls on the System Setup page. You can find definitions for the fields and controls later on this page.



System ID HRMS

Release Version 2.0

*Description HRMS

*Logging Level 3 - Information and above

*Temp Directory C:/TEMP/

Last Task Transaction ID 0

Last Appt. Transaction ID 436

☑ Allow Untrusted Certificate

Dynamic Display Branding

☑ Enable Top Branding Image

☐ Enable Bottom Branding Image

Upper Image Name NEW_PS_LOGO

Lower Image Name

Height 0

Width 0

File Transfer Options

Last Transaction ID 0

*File Chunk Size (Bytes) 500000

System ID

Enter a unique system ID. This ID must be unique across your organization to allow use across multiple data bases.

After being set, this ID is used internally as a key on different pieces of information.

For example, if you plan to connect to more than one PeopleSoft database, each database must have a unique ID. So if you have a Human Resources (HR) instance and a Financials/Supply Chain Management (FSCM) instance, each system should use a different system ID. When a transaction is passed to Microsoft Outlook, that transaction includes the system ID and the transaction ID from the database where it originated. Later, when the add-in needs to know the origin of that data, the information from the system ID and the transaction ID is used.

Release Version

The version of the system identified by the system ID. This number is predefined.

This number is used internally and is updated only when a new version of Desktop Integration is available.

Description

A description of the system that the system ID identifies. Enter a unique description.

The description can be a mixture of uppercase and lowercase with a maximum limit of 30 characters.

This description is visible within the Microsoft Office applications server configuration.

Logging Level

Specify the level of logging that you want. This controls the level of detail that is sent to the display log. Select a logging level from the drop-down list box. Values are:

- *0:* No logging

  This is the default. Due to minor performance impact, Oracle recommends that you select *0.*

- *1:* Errors
- *2:* Warnings and above
- *3:* Information and above

See Using the Display Log.

Last Task Transaction ID

The transaction ID for the last task that keeps the tasks synchronous. This ID is auto incremented and should not normally be adjusted.

Last Appt. Transaction ID (last appointment transaction identifier)

The date of the last appointment transaction.

This is autogenerated, tracked, and used for synchronization.

Last Transaction ID

This is autogenerated, tracked, and used for synchronization.

File Chunk Size (Bytes)

The file size with which to move data to a temporary directory. This value is predefined. Administrators can set up chunking size for performance purposes.

Temporary Directory

The temporary directory where you want to reassemble the data.

Files are spooled to a temporary directory and reassembled there before completing. This directory must be available on all application servers that may process the incoming Desktop Integration Broker messages.

See Using the Display Log.

Setting Up Tasks

This section discusses how to:

- Set up task registration.
- Set up the task queue.
- Compose task contents.

Pages Used to Set Up Tasks

Page Name	***Definition Name***	***Navigation***	***Usage***
Task Registration	EODI_TSK_REG	Enterprise Components, Desktop Integration, Task Registration	Select a task to be queued up and put into the user's tasks.
Task Queue - Task	EODI_TSK_QUEUE	Enterprise Components, Desktop Integration, Task Queue, Task	Define the things that you want as defaults in the user's tasks.
Task Queue - Message	EODI_TSK_MESSAGE	Enterprise Components, Desktop Integration, Task Queue, Message	Compose the message body of tasks.

Task Registration Page

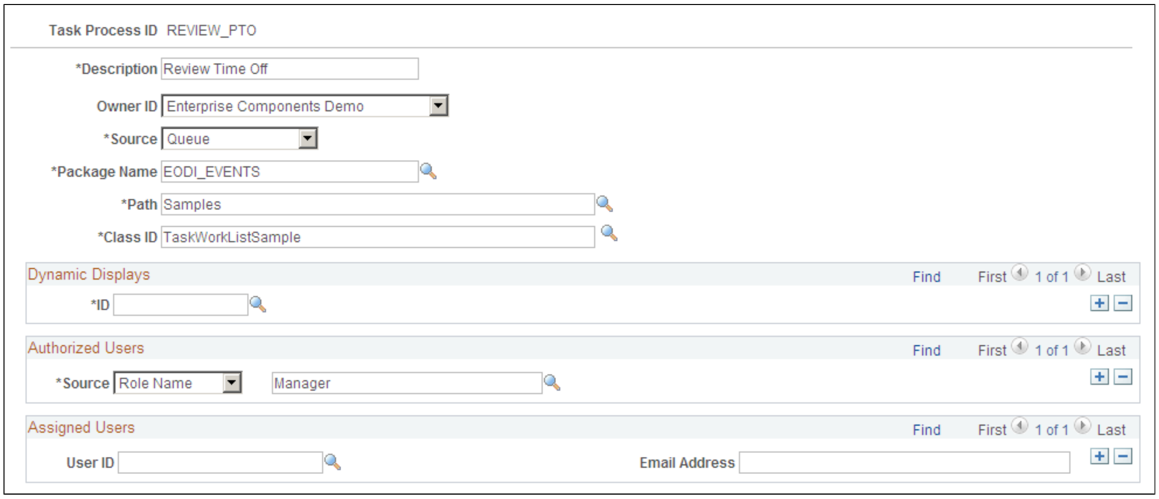
Use the Task Registration page (EODI_TSK_REG) to select a task to be queued up and put into the user's tasks.

Navigation

Enterprise Components, Desktop Integration, Task Registration

Image: Task Registration page

This example illustrates the fields and controls on the Task Registration page. You can find definitions for the fields and controls later on this page.



Task Process ID REVIEW_PTO

*Description: Review Time Off

Owner ID: Enterprise Components Demo

*Source: Queue

*Package Name: EODI_EVENTS

*Path: Samples

*Class ID: TaskWorkListSample

Dynamic Displays — Find First 1 of 1 Last

*ID:

Authorized Users — Find First 1 of 1 Last

*Source: Role Name — Manager

Assigned Users — Find First 1 of 1 Last

User ID: — Email Address:

In a generic, no customization implementation, use this page to define the Authorized Users section (source ID). All other fields are prepopulated and defined here for your reference only.

A task is a way for something to be queued up and put into the user’s tasks. Oracle delivers the tasks with the application. You can customize the tasks to suit your particular user groups and business needs.

The status of tasks is updated when you log into Microsoft Outlook.

Task Process ID

Displays the prepopulated ID, which is delivered with the application and is unique across your system.

You can create new IDs as a customization.

Description

Enter a text description of the task process.

Object Owner ID

Select the application that owns the object (for example, Enterprise Components). This drop-down list box that contains values for all of the available applications.

This is not a required field.

Source

Select the source of the task. Values are:

- *Queue*

  All task data is retrieved directly from the queue table. The PeopleSoft Application Layer for the particular process writes data into the queue tables. The data must be in the queue tables at the time of the request.

- *Application Class*

Task data is retrieved from the PeopleSoft Application Layer for the specific process. The process can retrieve its data from any available source within the PeopleSoft application. The data can be generated and retrieved at the time of the request.

Package Name

The name of the package that defines the PeopleSoft Application Layer for this process.

This is unique for each process. The Application Class information and data here is where you point to the code that will be called for your specific PeopleSoft Application Layer functions. This is needed only if you are modifying or adding new integrations. Applications deliver these defined for their delivered integrations.

The package name, path and class ID are specific to the registration. For delivered registrations, the fields are populated with application specific data.

When creating new registrations, these classes are defined beforehand in PeopleSoft Application Designer. This applies to all registrations.

Path

The path to the PeopleSoft Application Layer code. Enter the path where samples can be found.

Class ID

The class ID for the PeopleSoft Application Layer code. Enter a unique ID for the class.

Dynamic Displays

ID

Things that are supplied by default to the task. Select the ID of the display that you want as the default.

Authorized Users

Source

The source of the authorized users. Select from the drop-down list box. Values are:

- *User ID*
- *Role Name*
- *SQL User List:* An SQL user object that returns a list of operator IDs.

(the blank field to the right of the Source field)

Depending on the Source that you select in the Authorized User section (User ID, Role Name, SQL User List), a corresponding list of options becomes available for selection.

Assigned Users

User ID

This is generally assigned when the task is put into the queue. You can also assign by email selection.

You do not need to create a user ID for each; you can simply use the email address. The user ID does not need to match the email or PeopleSoft user ID. You can also create generic user IDs.

On the registration screens, these are the users that are assigned by default when something is added to the queue for this registration. For the actual processing, the assigned user on the actual queue item is used.

Email address

Alternatively, you can assign the Assigned User - User ID by email selection.

Task Queue - Task Page

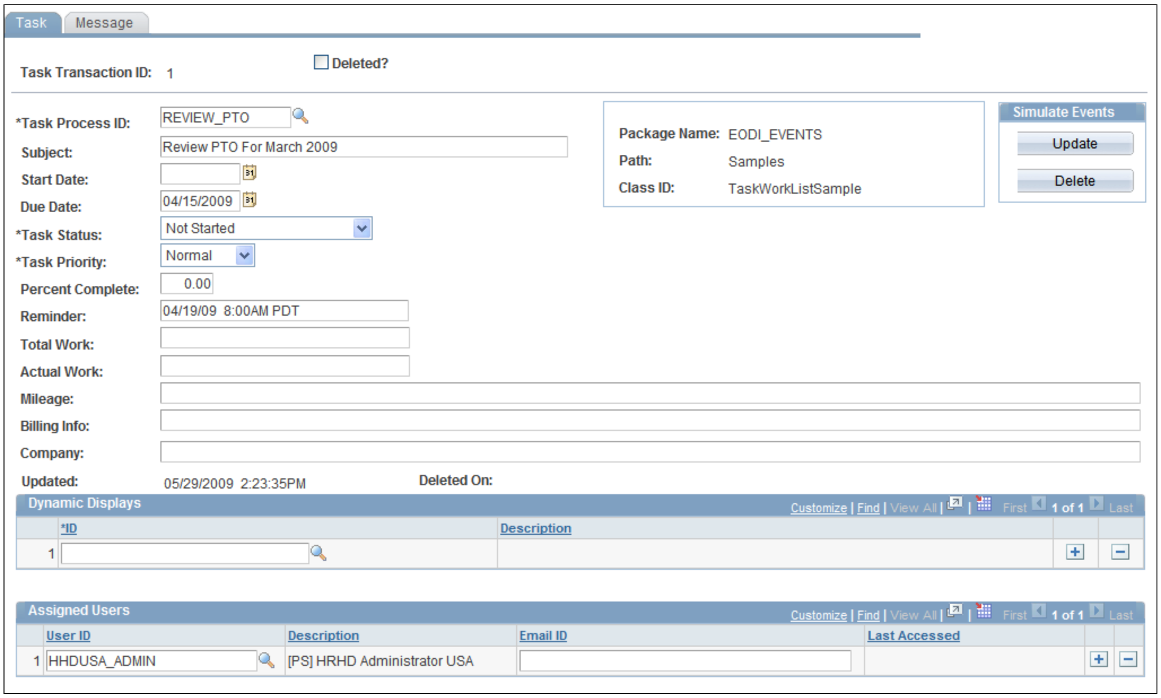
Use the Task Queue - Task page (EODI_TSK_QUEUE) to define the things that you want as defaults in the user's tasks.

Navigation

Enterprise Components, Desktop Integration, Task Queue, Task

Image: Task Queue - Task page

This example illustrates the fields and controls on the Task Queue - Task page. You can find definitions for the fields and controls later on this page.



Task Transaction ID:	1
□ Deleted?	
*Task Process ID:	REVIEW_PTO
Subject:	Review PTO For March 2009
Start Date:	
Due Date:	04/15/2009
*Task Status:	Not Started
*Task Priority:	Normal
Percent Complete:	0.00
Reminder:	04/19/09 8:00AM PDT
Total Work:	
Actual Work:	
Mileage:	
Billing Info:	
Company:	
Updated:	05/29/2009 2:23:35PM
Deleted On:	
Package Name:	EODI_EVENTS
Path:	Samples
Class ID:	TaskWorkListSample

Dynamic Displays

	*ID	Description
1		

Assigned Users

	User ID	Description	Email ID	Last Accessed
1	HHDUSA_ADMIN	[PS] HRHD Administrator USA		

Use this page to monitor task integrations or manually add or update tasks for users.

Generally, you do nothing in the queue. The queue is essentially a view into the tables as data is moved back and forth. You can use the queue to manually add tasks to a user's Microsoft Outlook task list; however, that would normally be handled by an application-specific feature (through the PeopleSoft Application Layer for a process).

Task Transaction ID

The ID is generated and supplied beforehand from the System Setup page. This is a unique ID for everything that is put in the Task Queue. Enter the transaction ID for the transaction that you want.

To autonumber, leave the transaction ID as zero, and it will be updated when you save.

See System Setup Page.

Deleted?

Select to flag for deletion.

Task Process ID

Displays the ID of the Task Process.

This is a prompt from the process ID for the registration. Select a Task Registration that you previously set up. The process ID is a 15–character, uppercase-only value. Enter the ID of the task process that you want.

Subject

Enter the task subject name.

All of the following fields correspond to the Microsoft Outlook fields that are displayed in a task. This is the information that is passed back and forth (updates in these fields are passed back and forth).

Start Date

Enter the date that you started the task or test.

Due Date

Enter the date that the task or test is due.

Task Status

Select the status or state of the task. The status is updated only when you log back in to Microsoft Outlook again. Values are:

- *Completed*
- *Deferred*
- *In Progress*
- *Not started*

  This is the default value.
- *Waiting on someone else.*

Task Priority

Select the priority that is defined for the task. Values are:

- *High*
- *Low*
- *Normal*

This is the default value.

Percent Complete

Enter the percentage of the task that has been completed.

Reminder

Enter any information or specifics that you want to recall, for example, the date and time to remind the user about this task.

This is the reminder that you can set within Microsoft Outlook.

For example, it might be an appointment reminder that pops up and sends you an email when a meeting is about to begin.

Total Work

Outlook task field.

This is a free form field that is available in Microsoft Outlook Task.

Actual Work

Outlook task field.

This is a free form field that is available in Microsoft Outlook Task.

Mileage

Outlook task field.

This is a free form field that is available in Microsoft Outlook Task.

Billing Info

Outlook task field.

This is a free form field that is available in Microsoft Outlook Task.

Company

Outlook task field.

This is a free form field that is available in Microsoft Outlook Task.

Updated

The date that this task information was last updated.

Deleted On

The date that this task was flagged for deletion. This field remains blank until or unless the task is flagged for deletion.

Dynamic Displays

ID

Select the ID of the dynamic display. Select any number of Dynamic Display IDs for the displays that you want to automatically show alongside a user’s tasks.

Description

Displays the description of the dynamic display ID, which is automatically supplied with the ID selection.

Simulate Events

Update	You can use the Simulate Events section to trigger PeopleCode without using Microsoft Outlook. Click the Update button to update and test your process, or select Delete? to flag this process for deletion. This feature enables development to test code without Microsoft Outlook. The PeopleSoft Application Layer application classes are called for the particular process.

Task Queue - Message Page

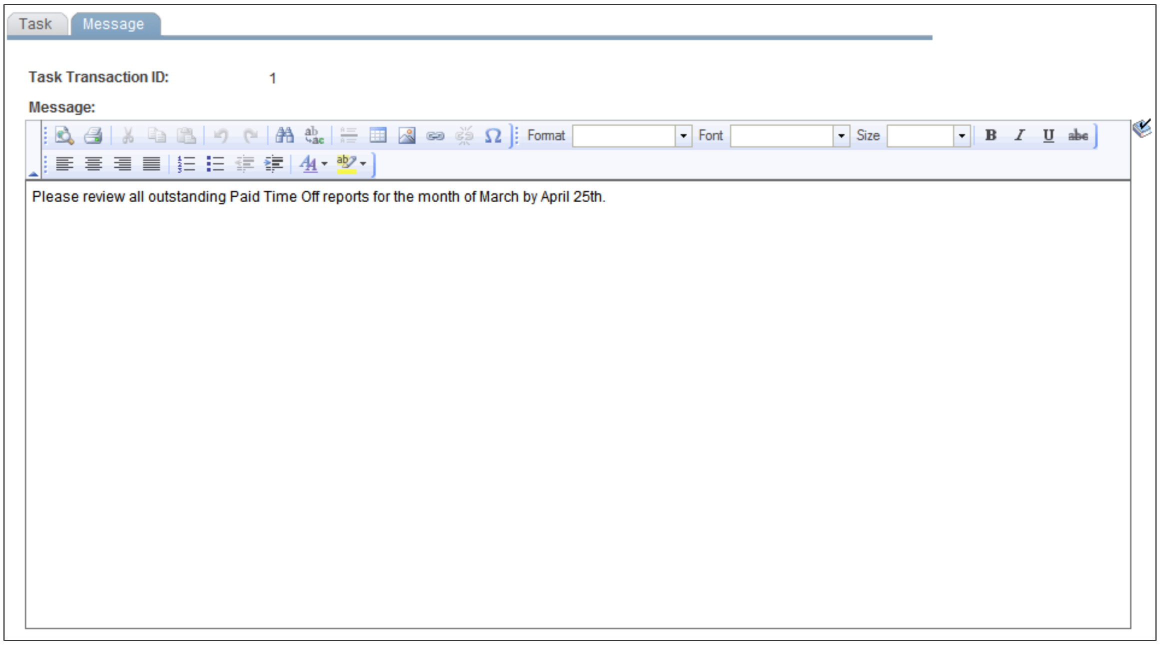
Use the Task Queue - Message page (EODI_TSK_MESSAGE) to compose the message body of tasks.

Navigation

Enterprise Components, Desktop Integration, Task Queue, Message

Image: Task Queue - Message page

This example illustrates the fields and controls on the Task Queue - Message page. You can find definitions for the fields and controls later on this page.



Task Transaction ID: 1

Message:

Please review all outstanding Paid Time Off reports for the month of March by April 25th.

Use this page to further define the body of the message. This is the message body of the task that is created in Microsoft Outlook and contains any changes that are made in Microsoft Outlook.

Registering Address Books

This section provides an overview of address book registration and discusses how to register your address book.

Page Used to Register Your Address Book

Page Name	***Definition Name***	***Navigation***	***Usage***
Address Book Registration	EODI_CON_REG	Enterprise Components, Desktop Integration, Address Book Registration	Register your address book.

Understanding Address Book Registration

Address books enable you to leverage PeopleSoft data to update Outlook users' local address lists or to control the email addresses that are available within the PeopleSoft Outlook Meeting Scheduler.

Several types of address books are available that you can define. The main difference between them is where the information originates. The type of address book determines which options are available in the registration page.

- *Internal address books*—These are derived from the Internal Address book page as defined within Desktop Integration.

  The address can be entered manually through the Internal Address Book page, or the underlying record can be updated by some other process.

- *Record/View address books*—These pull all of their information for a record or view as defined in PeopleTools and must conform to a specific structure.

  The record or view must contain the EODI_CON_SBR subrecord.

- *Application Class address books*—These pull all of their information from application classes and allow dynamic address books to be generated in PeopleCode.

Address Book Registration Page

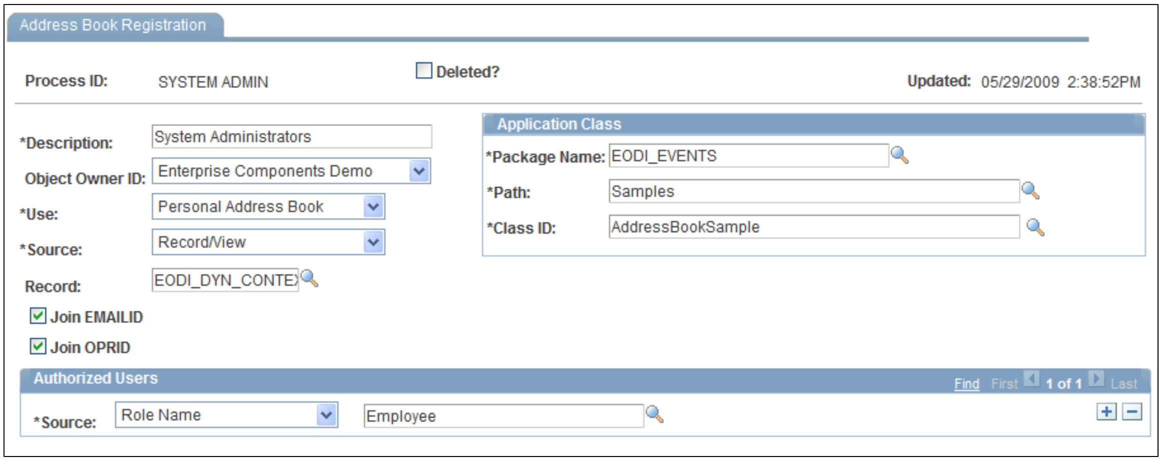
Use the Address Book Registration page (EODI_CON_REG) to register your address book.

Navigation

Enterprise Components, Desktop Integration, Address Book Registration

Image: Address Book Registration page

This example illustrates the fields and controls on the Address Book Registration page. You can find definitions for the fields and controls later on this page.



Address Book Registration

Process ID: SYSTEM ADMIN

□ Deleted?

Updated: 05/29/2009 2:38:52PM

*Description: System Administrators

Object Owner ID: Enterprise Components Demo

*Use: Personal Address Book

*Source: Record/View

Record: EODI_DYN_CONTE

☑ Join EMAILID

☑ Join OPRID

Application Class

*Package Name: EODI_EVENTS

*Path: Samples

*Class ID: AddressBookSample

Authorized Users

*Source: Role Name Employee

Use this page to register and define your available address book options.

Process ID This ID is defined for delivered integrations.

Deleted? Select to flag this address for deletion.

Updated This is automatically populated with the latest update saved to this address book.

Description Description of the address book. Enter a unique description.

Object Owner ID The application class that owns the object (for example, Enterprise Components).

Select a value from the drop-down list box that contains all of the available applications.

This is not a required field.

Use Specifies where the address book is used:

- *Personal Address Book:* Replicated to the users' Outlook Address Book list.
- *Meeting Invitees:* Only available if added to a meeting in the queue.

  The Meeting Invitees option enables two additional fields: Join EMAILID (join email identification) and Join OPRID (join operator identification).

Source The source of the address book or contact.

Values are:

- *Application Class*
- *Record/View*
- *Internal*

See Understanding Address Book Registration.

Record

Enter the source record. This field appears if the selected source is *Record/View*.

Join EMAILID

Select for Desktop Integration to automatically join the EMAILID field to the current users email address. This field appears if the selected source is *Record/View*.

The EMAILID field must be present on the record selected.

Join OPRID

Select for Desktop Integration to automatically join the OPRID field to the current users operator ID. This field appears if the selected source is *Record/View*.

The OPRID field must be present on the record selected.

Application Class

Package Name

The name of the package that defines the PeopleSoft Application Layer for this process.

This is unique for each process. The Application Class information and data here is where you point to the code that will be called for your specific PeopleSoft Application Layer functions. This is needed only if you are modifying or adding new integrations. PeopleSoft Applications deliver these defined for their delivered integrations.

The package name, path and class ID are specific to the registration. For delivered registrations, the fields are populated with application specific data.

When creating new registrations, these classes are defined beforehand in PeopleSoft Application Designer. This applies to all registrations.

Path

(Samples) Select the path where Samples can be found.

Class ID

Select a unique ID for the Class.

Authorized Users

Source

The source of the authorized user. Select from the drop-down list box.

Values are:

- *User ID*
- *Role Name*
- *SQL User List*

You do not need to create a user ID for each; you can simply use the email address. The user ID does not need to match the email or PeopleSoft user ID. You can also create generic user IDs.

The field to the right is automatically supplied based on the source that you specify.

(the field to the right of the Source field) Depending on the Source that you select in the Authorized User section (User ID, Role Name, SQL User List), a corresponding list of options becomes available for selection.

Registering Internal Address Books

This section discusses how to register and define your internal address book.

Page Used to Register Internal Address Books

Page Name	***Definition Name***	***Navigation***	***Usage***
Internal Address Book	EODI_CON	Enterprise Components, Desktop Integration, Internal Address Book	Register and define your available internal address book options.

Internal Address Book Page

Use the Internal Address Book page (EODI_CON) to register and define your available internal address book options.

Navigation

Enterprise Components, Desktop Integration, Internal Address Book

Internal address books are derived from the Internal Address book page as defined within Desktop Integration. The address can be entered manually through the Internal Address Book page, or the underlying record can be updated by some other process.

Address Book Process ID The ID is supplied as defined on the Address Book Registration page.

Internal address books are derived from the Internal Address book page as defined within Desktop Integration.

The address can be entered manually through the Internal Address Book page, or the underlying record can be updated by some other process.

Email Address The email address of the contact.

Deleted? Select to flag this email address or contact information for deletion.

Full Name The full name of the person or contact.

Title The title of the person or contact.

First Name The first name of the person or contact.

Middle Name The middle name of the person or contact.

Last Name The family or sir name of the person or contact.

Suffix The suffix for this person or contact.

Alt. Email (alternate email) An alternate email address to the primary listed email address.

Alt. Email (alternate email) An alternate email address to the secondary listed email address.

Company The company where this person or contact works.

Job Title The title of the position that this person or contact holds at the company.

Work Phone The work phone number of this person or contact.

Home Phone The home phone number of this person or contact.

Mobile Phone The mobile or cell phone number of this person or contact.

Updated This date is automatically updated with each saved revision to this address in the address book. This same date appears at the top of the page as well.

Setting Up Appointments

This section discusses how to:

- Set up appointment registration.
- Review the appointment queue.
- Set up meeting scheduling criteria.
- Set up meeting criteria.

Pages Used to Set Up Appointments

Page Name	***Definition Name***	***Navigation***	***Usage***
Appointment Registration	EODI_APT_REG	Enterprise Components, Desktop Integration, Appointment Registration	Set up appointment registration and to define he defaults for the user's appointments, invitations, and calendar.
Appointment/Meeting Group	EODI_APT_QUEUE	Enterprise Components, Desktop Integration, Appointment Queue, Appointment/Meeting Group	Review the appointment queue, monitor integration transactions, or manually add or update meetings and appointments.
Availability	EODI_APT_TIMES	Enterprise Components, Desktop Integration, Appointment Queue, Availability	Specify the system parameters for scheduling meetings.
Meeting	EODI_APT_MEETINGS	Enterprise Components, Desktop Integration, Appointment Queue, Meeting	Specify criteria for your meeting invitations and attendees.

Appointment Registration Page

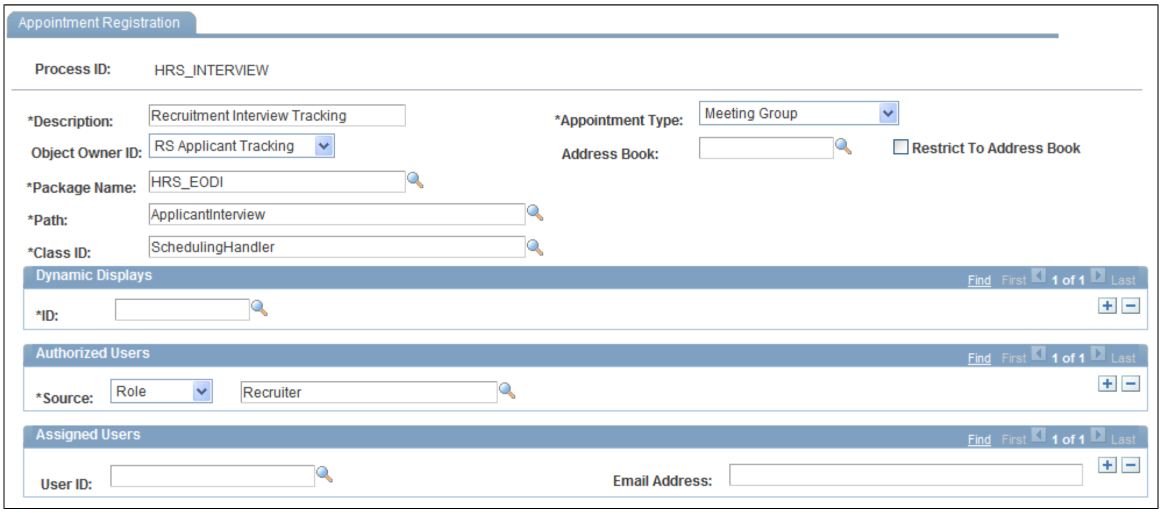
Use the Appointment Registration page (EODI_APT_REG) to set up appointment registration and to define he defaults for the user's appointments, invitations, and calendar.

Navigation

Enterprise Components, Desktop Integration, Appointment Registration

Image: Appointment Registration page

This example illustrates the fields and controls on the Appointment Registration page. You can find definitions for the fields and controls later on this page.



Appointment Registration

Process ID: HRS_INTERVIEW

*Description: Recruitment Interview Tracking

*Appointment Type: Meeting Group

Object Owner ID: RS Applicant Tracking

Address Book:

□ Restrict To Address Book

*Package Name: HRS_EODI

*Path: ApplicantInterview

*Class ID: SchedulingHandler

Dynamic Displays

*ID:

Authorized Users

*Source: Role Recruiter

Assigned Users

User ID:

Email Address:

This information is supplied by the applications for delivered integrations. You define the Authorized Users section here. As with tasks, if creating custom integrations, you create data here.

Process ID

The ID is supplied beforehand for delivered integrations.

Description

Description of the process. Enter a unique description.

Appointment Type

Type of appointment. Appointments are anything on a calendar. Values are:

- *Appointment:* No invites, just an item on a calendar.
- *Meeting:* Single, simple.
- *Meeting Group:* Schedule multiple simultaneously.

  For example, when scheduling an interview candidate, you have visibility to the entire series of interviews.

Object Owner ID

The application class that owns the object (for example, Enterprise Components).

Select a value from the drop-down list box that contains all of the available applications.

Address Book

Optionally, select a Desktop Integration address book to be available in the Outlook PeopleSoft Scheduler.

Restrict To Address Book

Select to restrict the available email address options to the address Book entered.

Package Name

The name of the package that defines the PeopleSoft Application Layer for this process. This is unique for each process.

The Application Class information and data here is where you point to the code that will be called for your specific PeopleSoft Application Layer functions. This is needed only if you are modifying or adding new integrations. PeopleSoft Applications deliver these defined for their delivered integrations.

The package name, path and class ID are specific to the registration. For delivered registrations, the fields are populated with application specific data.

When creating new registrations, these classes are defined beforehand in PeopleSoft Application Designer. This applies to all registrations.

Path

(Samples) Enter the path where Samples can be found.

Class ID

(Address Samples) Enter a unique ID for the class.

Dynamic Displays

ID

Things that are supplied by default to the appointment. Select the ID of the display that you want as the default.

Authorized Users

Source

The source of the authorized user. Select from the drop-down list box. Values are:

- *User ID*
- *Role Name*
- *SQL User List*

  Query from a list to obtain email addresses.

You do not need to create a user ID for each; you can simply use the email address. The user ID does not need to match the email or PeopleSoft user ID. You can also create generic user IDs.

(the blank field to the right of the Source field)

Depending on the Source that you select in the Authorized User section (User ID, Role Name, SQL User List), a corresponding list of options becomes available for selection.

Assigned Users

User ID

Generally assigned when the task is put into the queue. You can also assign by email selection.

Email Address

Alternatively, you can assign this field by email selection.

Appointment Queue - Appointment/Meeting Group Page

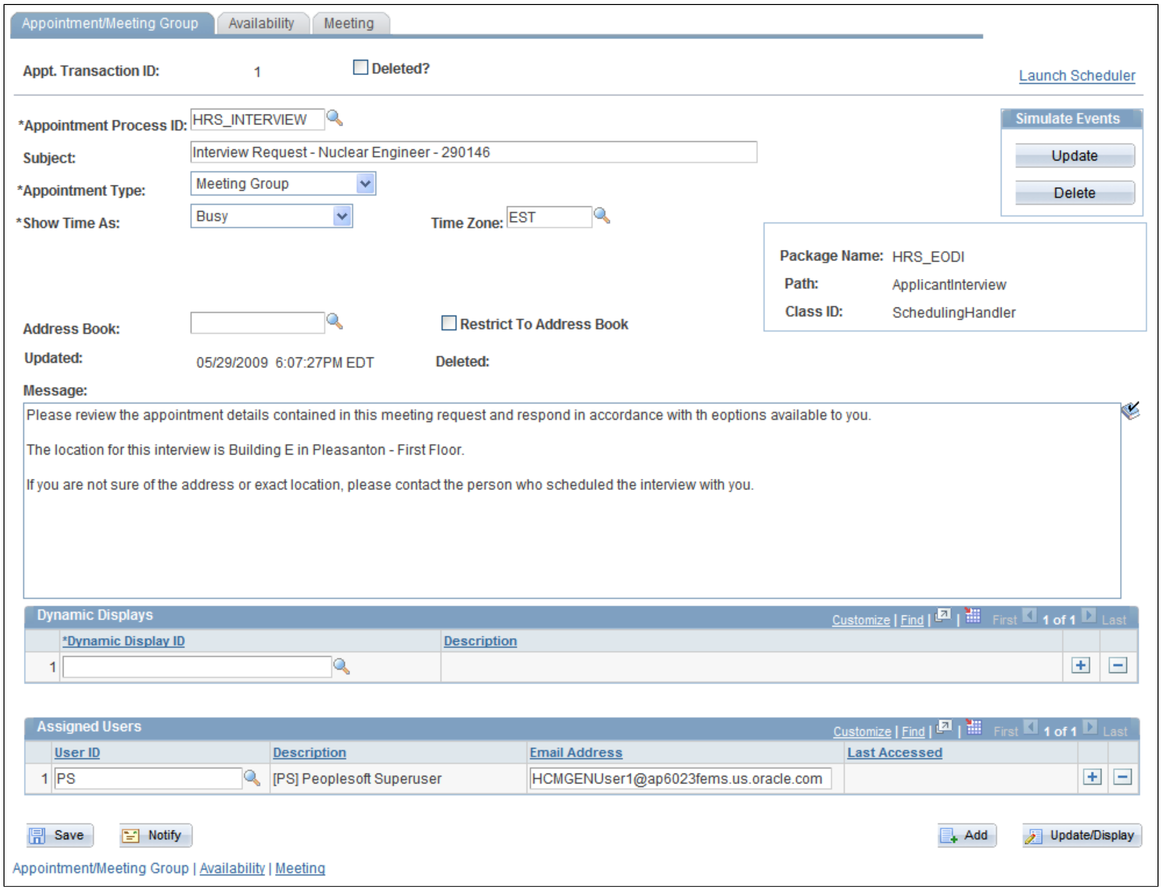
Use the Appointment Queue - Appointment/Meeting Group page (EODI_APT_QUEUE) to review the appointment queue, monitor integration transactions, or manually add or update meetings and appointments.

Navigation

Enterprise Components, Desktop Integration, Appointment Queue, Appointment/Meeting Group

Image: Appointment Queue - Appointment/Meeting Group page

This example illustrates the fields and controls on the Appointment Queue - Appointment/Meeting Group page. You can find definitions for the fields and controls later on this page.


Appointment/Meeting Group
Availability
Meeting
Launch Scheduler
Simulate Events
Update
Delete
Customize | Find | First 1 of 1 Last
Save
Notify
Add
Update/Display
Appointment/Meeting Group | Availability | Meeting

Appt. Transaction ID: 1

□ Deleted?

*Appointment Process ID: HRS_INTERVIEW

Subject: Interview Request - Nuclear Engineer - 290146

*Appointment Type: Meeting Group

*Show Time As: Busy

Time Zone: EST

Package Name: HRS_EODI

Path: ApplicantInterview

Class ID: SchedulingHandler

Address Book:

□ Restrict To Address Book

Updated: 05/29/2009 6:07:27PM EDT

Deleted:

Message:

Please review the appointment details contained in this meeting request and respond in accordance with th eoptions available to you.

The location for this interview is Building E in Pleasanton - First Floor.

If you are not sure of the address or exact location, please contact the person who scheduled the interview with you.

Dynamic Displays

	*Dynamic Display ID	Description
1		

Assigned Users

	User ID	Description	Email Address	Last Accessed
1	PS	[PS] Peoplesoft Superuser	HCMGENUser1@ap6023fems.us.oracle.com	

Appointment Process ID

The appointment process associated with this appointment or meeting.

Subject

- *Appointment:* The subject for the appointment in Outlook.
- *Meeting:* The description displayed in the Meeting Queue in the PeopleSoft Meeting Scheduler.

Appointment Type

- *Appointment:* Standard appointment.
- *Meeting:* Single meeting.
- *Meeting Group:* Multiple meetings to be scheduled on a single screen.

Show Time As

Status for the appointment. Values are:

- *Busy*

  This is the default value.

- *Free*
- *Out Of Office*
- *Tentative*

Start Date/Time

Start date and time for the appointment.

This is not visible for meetings.

End Date/Time

End date and time for the appointment.

This is not visible for meetings.

Address Book

The address book for the default on new meetings.

This is not available for appointments.

Restrict To Address Book

Option supplied by default onto new meetings.

Message

- *Appointments:* Body of appointment.
- *Meetings:* Master meeting body.

Dynamic Displays

Dynamic Displays ID

The ID of the dynamic display.

Select the ID of the display that you want as the default in the user's tasks.

Description

The description of the dynamic display that the ID identifies.

Assigned Users

User ID

This is generally assigned when the task is put into the queue.

You can also assign by email selection.

Appointment Queue - Availability Page

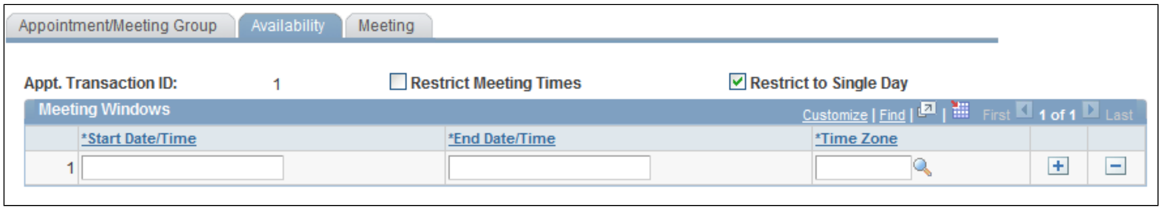
Use the Appointment Queue - Availability page (EODI_APT_TIMES) to specify the system parameters for scheduling meetings.

Navigation

Enterprise Components, Desktop Integration, Appointment Queue, Availability

Image: Appointment Queue - Availability page

This example illustrates the fields and controls on the Appointment Queue - Availability page. You can find definitions for the fields and controls later on this page.



Appt. Transaction ID: 1

□ Restrict Meeting Times

☑ Restrict to Single Day

Meeting Windows

	*Start Date/Time	*End Date/Time	*Time Zone		
1				+	−

Use this page to specify the system parameters that guide or restrict when you can schedule a meeting. This page is enabled for meetings only. The system considers meetings as a special type of appointment.

Restrict Meeting Times

When this option is enabled, meetings can be scheduled only within the meeting windows that you define.

Restrict to Single Day

When this option is enabled, all meetings in a meeting group must occur on the same day.

Meeting Windows

The start date and time, end date and time, and time zone.

Appointment Queue - Meeting Page

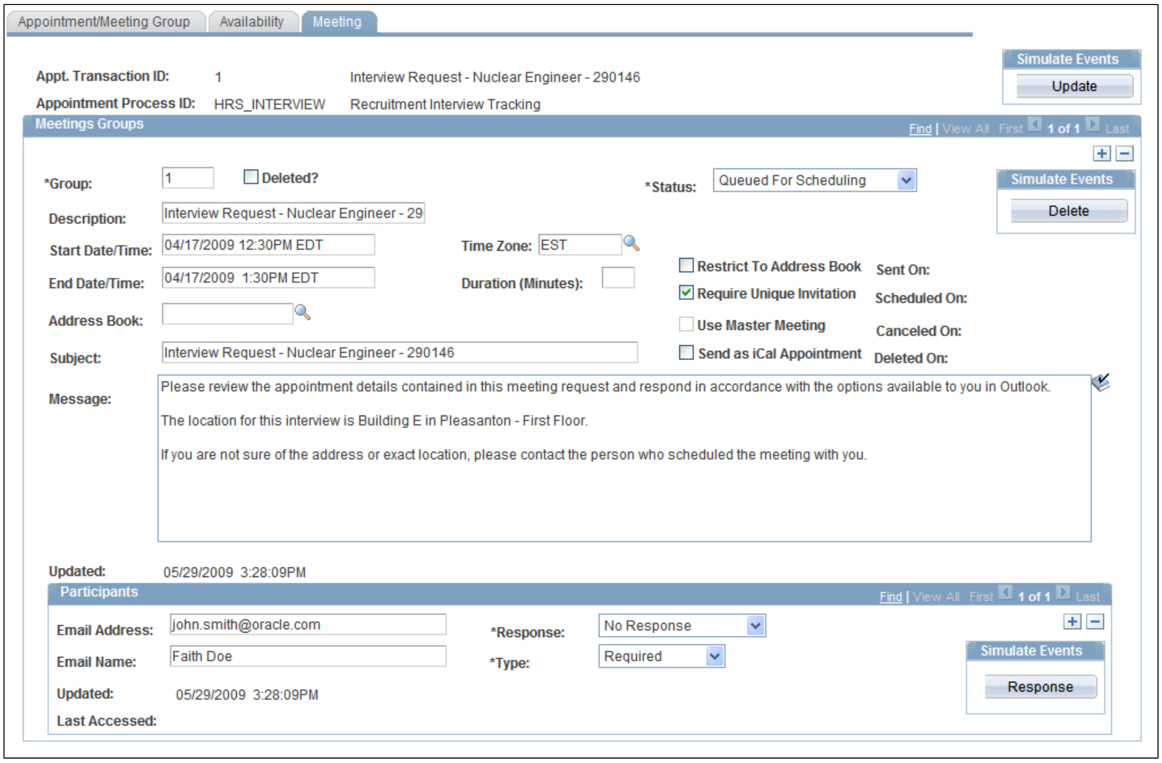
Use the Appointment Queue - Meeting page (EODI_APT_MEETINGS) to specify criteria for your meeting invitations and attendees.

Navigation

Enterprise Components, Desktop Integration, Appointment Queue, Meeting

Image: Appointment Queue - Meeting page

This example illustrates the fields and controls on the Appointment Queue - Meeting page. You can find definitions for the fields and controls later on this page.



Appt. Transaction ID: 1 Interview Request - Nuclear Engineer - 290146

Appointment Process ID: HRS_INTERVIEW Recruitment Interview Tracking

Meetings Groups

*Group: 1 □ Deleted? *Status: Queued For Scheduling

Description: Interview Request - Nuclear Engineer - 29

Start Date/Time: 04/17/2009 12:30PM EDT Time Zone: EST

End Date/Time: 04/17/2009 1:30PM EDT Duration (Minutes):

Address Book:

Subject: Interview Request - Nuclear Engineer - 290146

□ Restrict To Address Book Sent On:

☑ Require Unique Invitation Scheduled On:

□ Use Master Meeting Canceled On:

□ Send as iCal Appointment Deleted On:

Message: Please review the appointment details contained in this meeting request and respond in accordance with the options available to you in Outlook.

The location for this interview is Building E in Pleasanton - First Floor.

If you are not sure of the address or exact location, please contact the person who scheduled the meeting with you.

Updated: 05/29/2009 3:28:09PM

Participants

Email Address: john.smith@oracle.com *Response: No Response

Email Name: Faith Doe *Type: Required

Updated: 05/29/2009 3:28:09PM

Last Accessed:

This page is enabled for meetings only.

Update

In the Simulated Events section, click the Update button to test your integration process without using Microsoft Outlook.

This process updates all meeting groups.

Delete

In the Simulated Events section, click the Delete button to test your integration process without using Microsoft Outlook.

This process deletes the current meeting.

Status

Select a value from the drop-down list box. Values are:

- Awaiting Responses.
- Canceled.
- Initial - Not Visible.

  This is the default value.
- On Hold - Do Not Schedule.

- Queued for Canceling.
- Queued for Scheduling.
- Scheduled.

When you install the Microsoft Outlook Add-In program, the Status field is visible in your PeopleSoft application.

Start Date/Time

Enter the first date that you want to consider within a particular time frame or known availability.

From the drop-down list box, select the time that you want the meeting to begin.

Time Zone

The attendee's time zone.

When you select a time zone from the drop-down list box, you are ensuring that the time you enter on your calendar appears correctly on the attendee's invitation and calendar for his or her time zone.

End Date/Time

Enter the last date that you want to consider within a particular time frame or known availability.

From the drop-down list box, select the time that you want the meeting to end. This field is automatically populated when you specify a value in the Duration field.

Duration (Minutes)

Duration of the meeting time in minutes. This field is automatically populated when you select the beginning and ending time.

Address Book

Select from your available Desktop Integration address books.

Selecting a Desktop Integration address book enables you to make available an email address book that is sourced from the PeopleSoft application.

Restrict To Address Book

When you select this option, you can choose addresses only from the selected address book.

Subject

Enter a subject line. This can be the topic of the meeting, the call-in number, or whatever information you want.

Require Unique Invitation

Select to require a unique or custom invitation to a particular invitee or group.

This option prevents the user from sending the Master Meeting invite information in the Outlook Scheduler.

Note: Within the PeopleSoft Outlook Meeting Scheduler, the user can define a single meeting invite message for multiple meetings, individual meeting invite messages, or a combination of the two. The Master Meeting is a shared meeting invite that can be used and it's body is supplied by default from the Message as defined on the Appointment/Meeting Group page. Each individual meeting can also have a message body that is defined on the Meeting page. This text is used if the user chooses not to use the Master Meeting invite, or if it's use is restricted.

Use Master Meeting

Select to use the content or text specified in the Master Meeting for everyone to whom you are sending an invitation.

Send as iCal Appointment (iCalendar appointment)

Select to send to an external invitee who is using another calendaring system or application.

Deleted?

Select to flag this meeting for deletion.

Message

Enter the body of the invitation content or text into this pane for global visibility, unless otherwise specified.

The system uses this message if the meeting is not using the Master Meeting (if you did not select the Master Meeting check box).

Updated

This field is automatically populated with the date of the latest update.

Email Address

In the Participants section, enter the email address or D-list of the people you want to invite.

Email Name

Alternatively, you can enter the invitee name or group name.

Select from the list of available types and participant types:

- *Required:* User must be present at meeting.
- *Optional:* User can optionally attend the meeting.
- *Resource:* Used for room scheduling and resources.

Response

This is the current response received from the user. Values are:

- *Accepted*
- *Declined*
- *Deleted*
- *No Response*

  The invitation has been sent but no reply has been received.
- *Not Sent*

- *Proposed New Time*
- *Tentative*

Simulate Events

Response

In the Simulate Events section, click the Response button to simulate the event and test your process without using Microsoft Outlook.

Setting Up File Transfers

This section discusses how to:

- Set up file transfer registration.
- Review the file transfer queue.

Pages Used to Set Up File Transfers

Page Name	***Definition Name***	***Navigation***	***Usage***
File Transfer Registration	EODI_XFR_REG	Enterprise Components, Desktop Integration, File Transfer	Set up file transfer registration.
File Transfer Queue	EODI_XFR	Enterprise Components, Desktop Integration, File Transfer Queue	Review what you previously defined as defaults in your file transfers.

File Transfer Registration Page

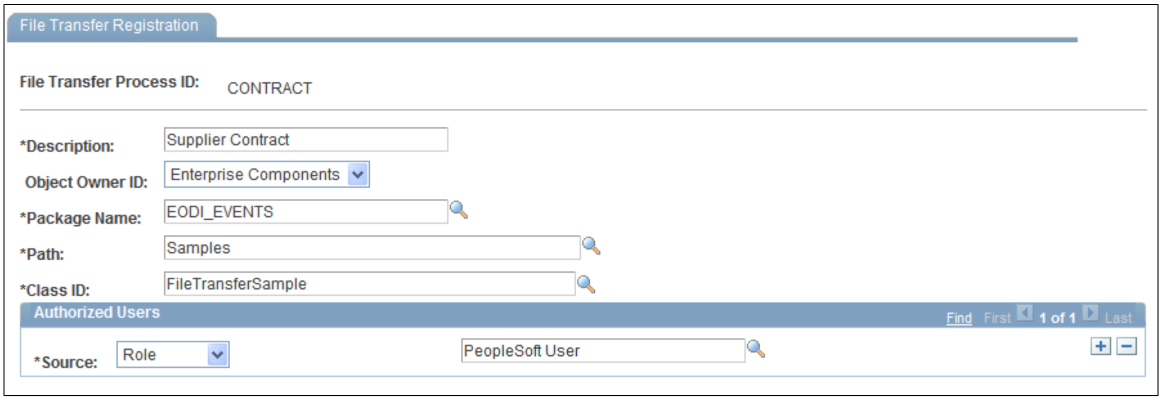
Use the File Transfer Queue page (EODI_XFR) to review what you previously defined as defaults in your file transfers.

Navigation

Enterprise Components, Desktop Integration, File Transfer Queue

Image: File Transfer Registration page

This example illustrates the fields and controls on the File Transfer Registration page. You can find definitions for the fields and controls later on this page.



File Transfer Registration

File Transfer Process ID: CONTRACT

*Description: Supplier Contract

Object Owner ID: Enterprise Components

*Package Name: EODI_EVENTS

*Path: Samples

*Class ID: FileTransferSample

Authorized Users

*Source: Role PeopleSoft User

File Transfer Process ID

The ID is supplied beforehand.

Description

Description of the File Process. Enter a unique description.

Object Owner ID

The application class that owns the object (for example, Enterprise Components).

Select a value from the drop-down list box that contains all of the available applications.

Package Name

The name of the package that defines the PeopleSoft Application Layer for this process. This is unique for each process.

The Application Class information and data here is where you point to the code that will be called for your specific PeopleSoft Application Layer functions. This is needed only if you are modifying or adding new integrations. PeopleSoft Applications deliver these defined for their delivered integrations.

The package name, path and class ID are specific to the registration. For delivered registrations, the fields are populated with application specific data.

When creating new registrations, these classes are defined beforehand in PeopleSoft Application Designer. This applies to all registrations.

Path

(Samples) Enter the path where Samples can be found.

Class ID

(Task Samples) Enter a unique ID for the class.

Authorized Users

Source

The source of the authorized user. Select from the drop-down list box. Values are:

- *User ID*
- *Role Name*
- *SQL User List*

You do not need to create a user ID for each; you can simply use the email address. The user ID does not need to match the email or PeopleSoft user ID. You can also create generic user IDs.

The field to the right is automatically populated based on the source that you specify.

(the blank field to the right of the Source field)

Depending on the Source that you select in the Authorized User section (User ID, Role Name, SQL User List), a corresponding list of options becomes available for selection.

File Transfer Queue Page

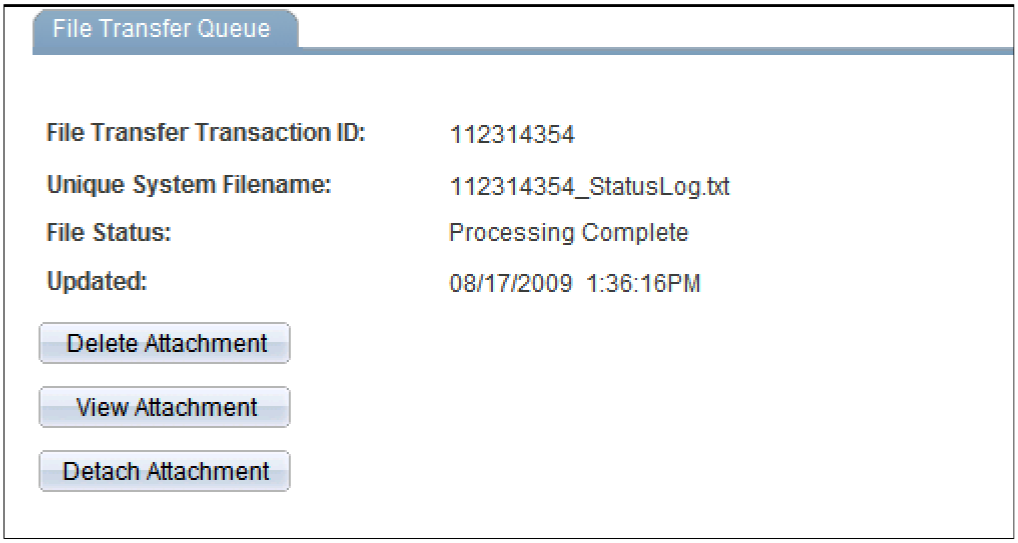
Use the File Transfer Queue page (EODI_XFR) to review what you previously defined as defaults in your file transfers.

Navigation

Enterprise Components, Desktop Integration, File Transfer Queue

Image: File Transfer Queue page

This example illustrates the fields and controls on the File Transfer Queue page. You can find definitions for the fields and controls later on this page.



File Transfer Transaction ID:	112314354
Unique System Filename:	112314354_StatusLog.txt
File Status:	Processing Complete
Updated:	08/17/2009 1:36:16PM

File Transfer Transaction ID

The numeric transaction ID that the PeopleSoft system assigns when you create the file transfer.

Unique System Filename

Upon processing of the file transfer, this field updates with the unique system file name assigned by the PeopleSoft system.

File Status

Displays the current status of this file transfer. Values are:

- Receiving File
- Assembling File
- Processing Application Logic
- Processing Complete

Updated

The date that this file transfer is updated with new or edited information.

Delete Attachment

Click to delete the attached file.

View Attachment

Click to view the attached file.

Detach Attachment

Click to detach the attached file.

Defining Dynamic Displays

This section provides an overview of dynamic displays and discusses how to:

- Define general dynamic display options.
- Define field and layout options for dynamic displays.

Pages Used to Define Dynamic Displays

Page Name	*Definition Name*	*Navigation*	*Usage*
Display Definition	EODI_DYN	Enterprise Components, Desktop Integration, Dynamic Display Definition, Display Definition	Define general dynamic display options.
Fields and Layout	EODI_DYN_FIELDS	Enterprise Components, Desktop Integration, Dynamic Display Definition, Fields and Layout	Define field and layout options for dynamic displays.

Understanding Dynamic Displays

Dynamic Displays allow the display of information from your PeopleSoft system alongside items in Microsoft Office. You can use Dynamic Displays to show any information contained within the PeopleSoft system and those displays can optionally be shown automatically alongside the Microsoft

Office items. The information contained within a Dynamic Display can also be driven contextually by the current item. For example, customer information from the PeopleSoft system can be displayed alongside an inbound customer email, or contact in an address book.

Display Definition Page

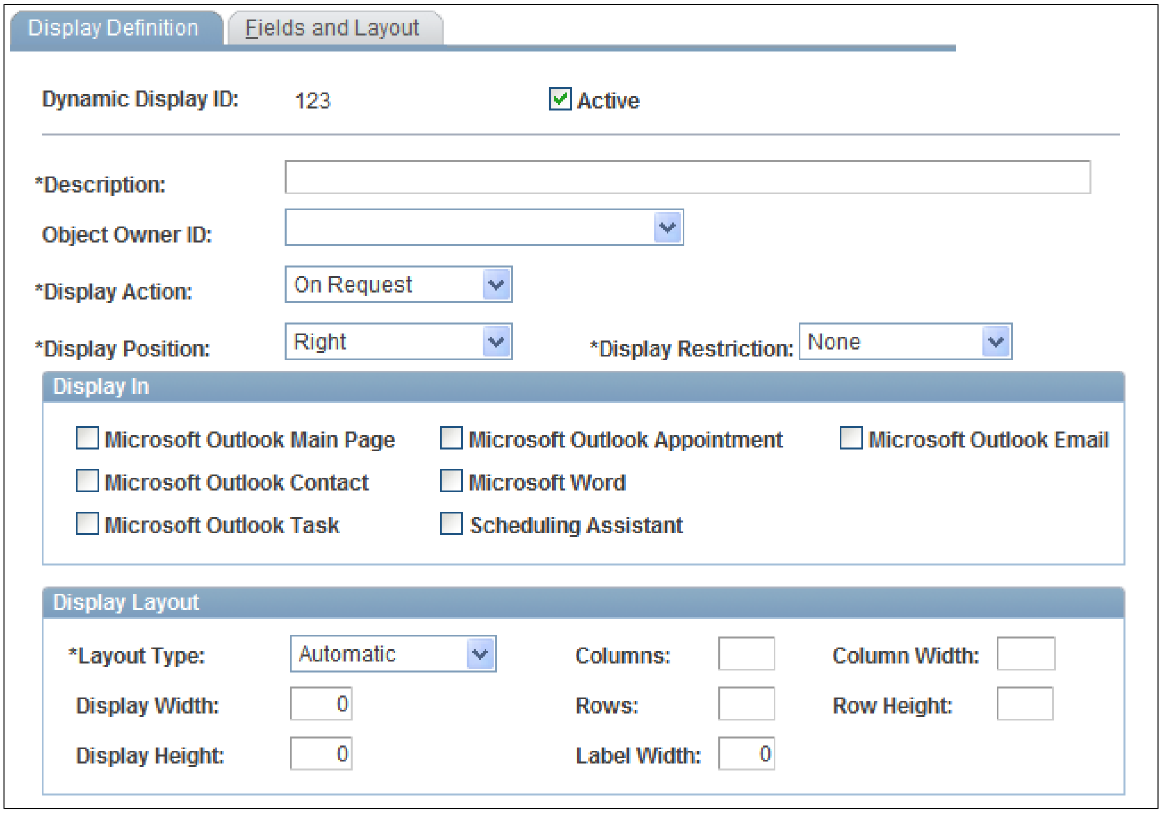
Use the Display Definition page (EODI_DYN) to define general dynamic display options.

Navigation

Enterprise Components, Desktop Integration, Dynamic Display Definition, Display Definition

Image: Display Definition page (1 of 2)

This example illustrates the fields and controls on the Display Definition page (1 of 2). You can find definitions for the fields and controls later on this page.



Dynamic Display ID: 123 ☑ Active

*Description:

Object Owner ID:

*Display Action: On Request

*Display Position: Right *Display Restriction: None

Display In

☐ Microsoft Outlook Main Page ☐ Microsoft Outlook Appointment ☐ Microsoft Outlook Email

☐ Microsoft Outlook Contact ☐ Microsoft Word

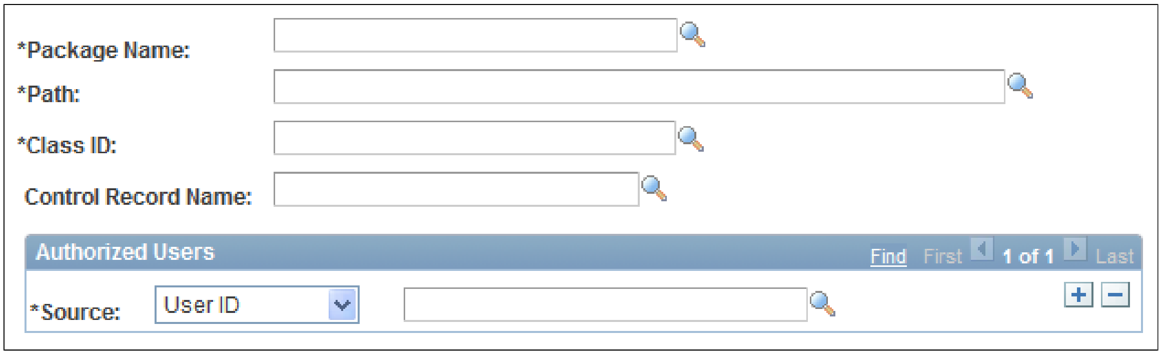
☐ Microsoft Outlook Task ☐ Scheduling Assistant

Display Layout

*Layout Type:	Automatic	Columns:		Column Width:	
Display Width:	0	Rows:		Row Height:	
Display Height:	0	Label Width:	0		

Image: Display Definition page (2 of 2)

This example illustrates the fields and controls on the Display Definition page (2 of 2). You can find definitions for the fields and controls later on this page.



*Package Name:

*Path:

*Class ID:

Control Record Name:

Authorized Users

*Source: User ID

Dynamic Displays dynamically set up the form and format with the required action fields, and then pass the information back to the form.

Dynamic Display ID The ID of the dynamic display is supplied as defined earlier. This must be a unique ID across your organization.

Description

The name of the dynamic display. Enter a unique name for this display.

Display Action

Select the display action. Values are:

- *Automatic*: The display is automatically shown if the transaction requests.
- *On Request*: The user must use the Display ribbon action item to display.

  You must use the Display ribbon action or menu item to display.

Display Position

Select the initial position of the Dynamic Display. Values are:

- *Right*
- *Left*
- *Top*
- *Bottom*
- *Floating*

Display Restriction

Used to restrict the position change of the Dynamic Display. This is useful if the display layout does not function well in a particular mode.

- *None*
- *No Horizontal*: Display restricted to Right, Left or Floating.
- *No Vertical*: Display restricted to Top, Bottom or Floating.
- *Frozen*: Display cannot be moved.

Display In

Select the Microsoft Office application areas where the Dynamic Display is available. Values are:

- Microsoft Outlook Main Page: The main frame of Microsoft Outlook; not tied to any single Microsoft Outlook item.
- Microsoft Outlook Contact
- Microsoft Outlook Task
- Microsoft Outlook Appointment: Includes any appointment, including meetings.
- Microsoft Word
- Scheduling Assistant: The PeopleSoft Scheduling Assistant, used for PeopleSoft meetings.
- Microsoft Outlook Email

When you select Microsoft Outlook Email, *Email Mode* becomes available for optional selection.

- *Email Mode*: Controls which email modes the Dynamic Display is available in.

  Values are:

  - *Read and Compose*
  - *Read*
  - *Compose*

Layout Type

Controls the layout method for the Dynamic Display. Values are:

- *Automatic*: Display layout is computed automatically based on the Row and Column assigned to individual controls (recommended).
- *Custom*: Display layout is precisely controlled by the X and Y layout positions.

Display Width

Width of the Dynamic Display. This will be ignored for Top or Bottom Display Positions.

If a display is a LEFT or RIGHT display position, then it will *always* go from top to bottom of the current page (but you can control the width).

Display Height

Height of the Dynamic Display. This will be ignored for Top or Bottom Display Positions.

If a display is a TOP or BOTTOM display position, it will always go the entire width of the current page (but you can control the height).

Columns

Number of columns for the display layout.

Column Width

Width of columns for the display layout.

Rows

Number of rows for the display layout.

Row Height

Height of rows for the display layout.

Package Name

The name of the package that defines the PeopleSoft Application Layer for this process. This is unique for each process.

The Application Class information and data here is where you point to the code that will be called for your specific PeopleSoft Application Layer functions. This is needed only if you are modifying or adding new integrations. PeopleSoft Applications deliver these defined for their delivered integrations.

The package name, path and class ID are specific to the registration. For delivered registrations, the fields are populated with application specific data.

When creating new registrations, these classes are defined beforehand in PeopleSoft Application Designer. This applies to all registrations.

Path

The path of the dynamic display sample. Select the path where the dynamic display samples can be found.

Class ID

Select a unique ID for the class.

Control Record Name

Optional record name. Fields from this record name can be defined as defaults in the Fields and Layout information to quickly build a Dynamic Display.

The properties of the record fields are also used.

Authorized Users

Source

The source of the authorized user. Select from the drop-down list box. Values are:

- *User ID*
- *Role Name*
- *SQL User List*

You do not need to create a user ID for each; you can simply use the email address. The user ID does not need to match the email or PeopleSoft user ID. You can also create generic user IDs.

The field to the right is automatically populated based on the source that you specify.

(the blank field to the right of the Source field)

Depending on the Source that you select in the Authorized User section (User ID, Role Name, SQL User List), a corresponding list of options becomes available for selection.

Fields and Layout Page

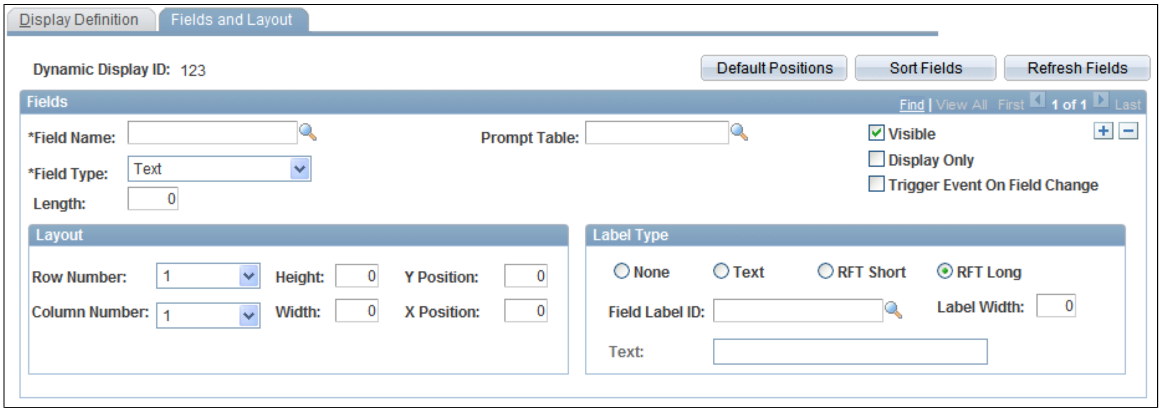
Use the Fields and Layout page (EODI_DYN_FIELDS) to define field and layout options for dynamic displays.

Navigation

Enterprise Components, Desktop Integration, Dynamic Display Definition, Fields and Layout

Image: Dynamic Displays - Fields and Layout page

This example illustrates the fields and controls on the Dynamic Displays - Fields and Layout page. You can find definitions for the fields and controls later on this page.



Dynamic Display ID: 123

*Field Name:

Prompt Table:

☑ Visible

□ Display Only

□ Trigger Event On Field Change

*Field Type: Text

Length: 0

Row Number: 1

Height: 0

Y Position: 0

Column Number: 1

Width: 0

X Position: 0

○ None ○ Text ○ RFT Short ● RFT Long

Field Label ID:

Label Width: 0

Text:

Default Position

Click this button to set the default positions for the Row and Column numbers of all fields on the page. The system uses the row height and column width from the main page to calculate the X and Y position of the fields. This button is only visible when you select the *Custom* layout type on the Display Definition page.

See Display Definition Page.

Sort Fields

Click to sort the fields within the scroll by Row and Column number.

Refresh Fields

Click to refresh all fields within the scroll with the fields from the Control record defined on the Display Definition page.

Field Name

Enter or select a unique name (must be unique within this display). All code references to the field will use this name.

Field Type

Select a value from the drop-down list box: Values are:

- Button
- Chart
- Date
- Drop-down List Box
- Grid
- Hyperlink
- Number
- Text

- Tree

Length

Enter the length of the input data. This field is valid for *Text* and *Number* field types only.

Prompt Table

Select a prompt table record name for the text field. Only valid for *Text* field types.

When you select a Prompt Table record name for Text field types, a Prompt Table edit check box becomes available. This option asks *Should Prompt Table be used as an edit table?*

This is only valid for *Text* field types where a *Prompt Table* has been defined.

Visible

Select to define the control as visible.

Display Only

Select to define the control as visible in the display only.

Trigger Event On Field Change

Select to trigger a field change event on the server if the current field is changed.

Decimal Positions

Enter the number of decimal positions in the number. This field is valid and enabled for *Number* field types only.

Button Action

Select the action to be performed when the button is clicked. This field is valid and enabled for *Button* field types only.

Values are:

- Capture Document Contents
- Capture Document Selection
- File/Document Download
- File/Document Upload
- Insert Into Current Document
- None

Layout

Row Number

Enter the number of rows for automatic layout.

Instead, you can click the Default Positions button to set the default positions for the Row and Column numbers of all fields on the page. The system uses the row height and column width from the main page to calculate the X and Y position of the fields. This button is only visible when you select the *Custom* layout type on the Display Definition page.

Column Number

Enter the number of columns for automatic layout.

Instead, you can click the Default Positions button to set the default positions for the Row and Column numbers of all fields on the page. The system uses the row height and column width from the main page to calculate the X and Y position of the fields. This button is only visible when you select the *Custom* layout type on the Display Definition page.

Height

Enter the height of the control. If you leave this value as *0,* the system uses the current row height as the default.

Width

Enter the height of the control.

If this is a *Display Only* field and you leave the value as *0,* the system will display the entire item.

If this is an *Input* field, you must specify the display *Width* of the input field.

X Position

Specify the absolute position for this control. This field is only visible when you select the *Custom* layout type on the Display Definition page.

Y Position

Specify the absolute position for this control. This field is only visible when you select the *Custom* layout type on the Display Definition page.

Label Type

None, Text, RTF Short or RTF Long

Select one of these radio buttons as the label type:

- None
- Text: Uses the hardcoded text that is supplied.
- RTF Short: Looks up the internal short text for the field, based on the supplied Field Label.
- RTF Long: Looks up the internal long text for the field, based on the supplied Field Label.

RTF Long and RTF Short are only valid options if the field name provided exists as a valid field without PeopleSoft Application Designer.

Field Label ID

When you select either *RTF Short* or *RTF Long* for the label type, specify the Field Label ID.

Field Label ID is only a valid option if the field name provided exists as a valid field without PeopleSoft Application Designer.

Label Width

Specify the width reserved for the label for this item. To line up all items on a single Dynamic Display, keep this number consistent across all fields.

Text	When you select Text for the Label Type, specify the label text.

Using the Display Log

This section discusses how to use the display log.

Page Used to View the Display Log

Page Name	***Definition Name***	***Navigation***	***Usage***
Desktop Integration Log	EODI_LOG	Enterprise Components, Desktop Integration, Display Log	View dynamic display log messages.

Desktop Integration Log Page

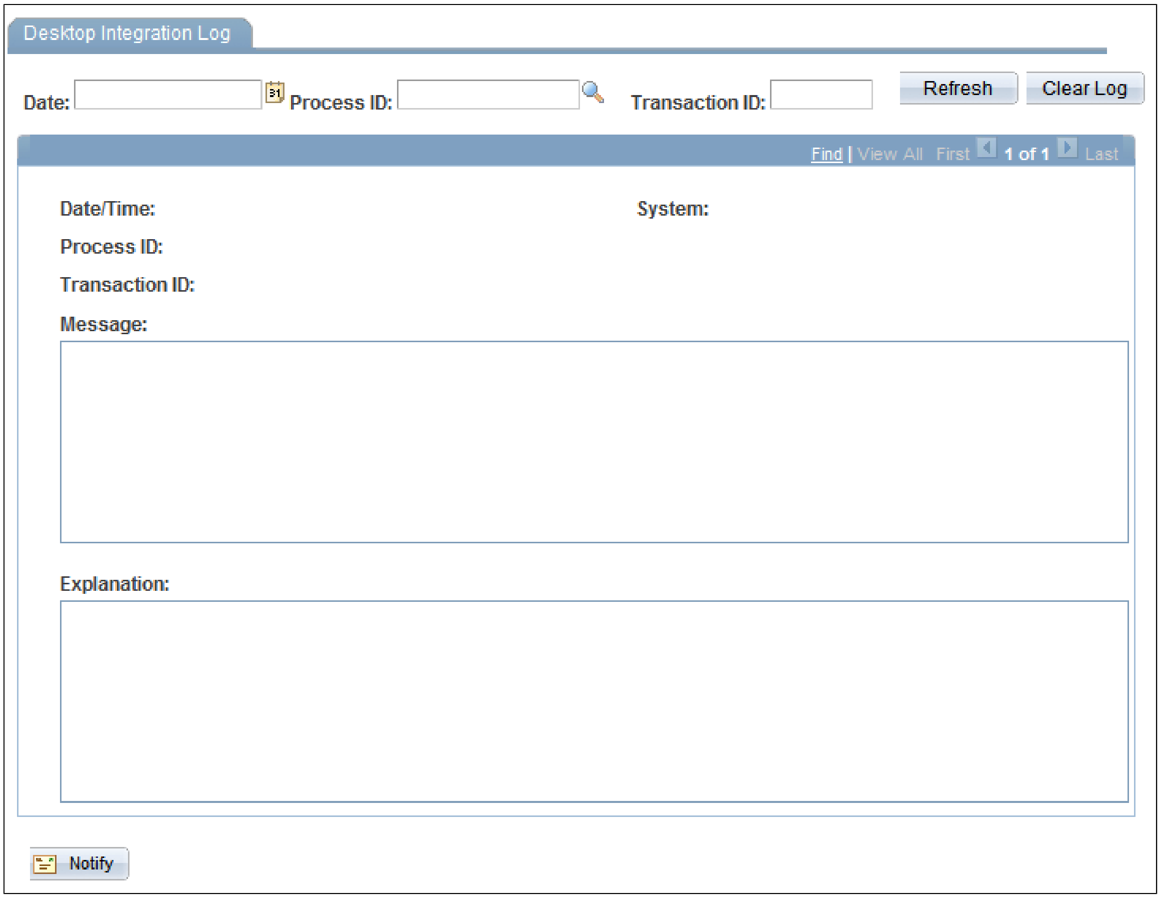
Use the Desktop Integration Log page (EODI_LOG) to view dynamic display log messages.

Navigation

Enterprise Components, Desktop Integration, Display Log

Image: Desktop Integration Log page

This example illustrates the fields and controls on the Desktop Integration Log page. You can find definitions for the fields and controls later on this page.



The Desktop Integration Log helps you understand and troubleshoot the Desktop Integration processes. The amount of detail shown in the log is determined by the log level set in the System Setup. As Desktop Integration processes run, information is written to the log at each step.

The Desktop Integration Log enables you to select what you want to view, and then displays the details of that selection for the messages that are passed. All specified information is optional and only filters the log that is retrieved when you click the Refresh button.

Date

Select the date for which you want to view the log entries and events.

Process ID

Select the ID of the process for which you want to view the log entries and events.

Transaction ID

Enter the ID of the transaction for which you want to view the log entries and events.

Refresh

Click to refresh the log entries and events that appear.

Clear Log	Click to clear the log entries and events that appear.

Log

Date/Time	The date and time of the log entry and event.
System	The system for which the log entry and event are associated.
Process ID	The unique ID of the process.
Transaction ID	The ID of the transaction of the log entry and event.
Message	The message in the log entry and event.
Explanation	The explanation of the log entry and event.

Using the PeopleSoft Outlook Menu

This section provides an overview of the PeopleSoft and Outlook integration, and discusses how to:

- Define settings and options.
- Use the PeopleSoft Scheduler and Scheduling Assistant.
- Replicate appointments, meetings, tasks, and address books.

Understanding the PeopleSoft and Outlook Integration

Within Microsoft Outlook, the following two components are installed that enable all of the communication and interaction between Microsoft Outlook and the PeopleSoft system:

- The Microsoft Add-In—this is represented in the PeopleSoft Outlook menu and handles all of the communication and information within Microsoft Outlook.
- The PeopleSoft Outlook URL Handler—this is an external Windows application that allows you to drill into Microsoft Outlook directly from the PeopleSoft system.

Defining Settings and Options

When you select Settings & Options from the PeopleSoft menu in Microsoft Outlook, the following pages are available:

- Servers page—shows the server detail (such as the server URL, connectivity, and the server functions that you selected).
- Options page—shows the display options and frequency of updates.
- Status Log page—shows the send and receive data and errors in the log.

Note: After the Microsoft Outlook Add-In installation completes, you must go to the Settings and Options - Servers page (PeopleSoft Outlook menu, Settings & Options, Servers) and define the server URL and functions.
The server must show the Enable check box selected to attempt connectivity. If the server was determined to be online during the last connection attempt, the Online check box is selected.

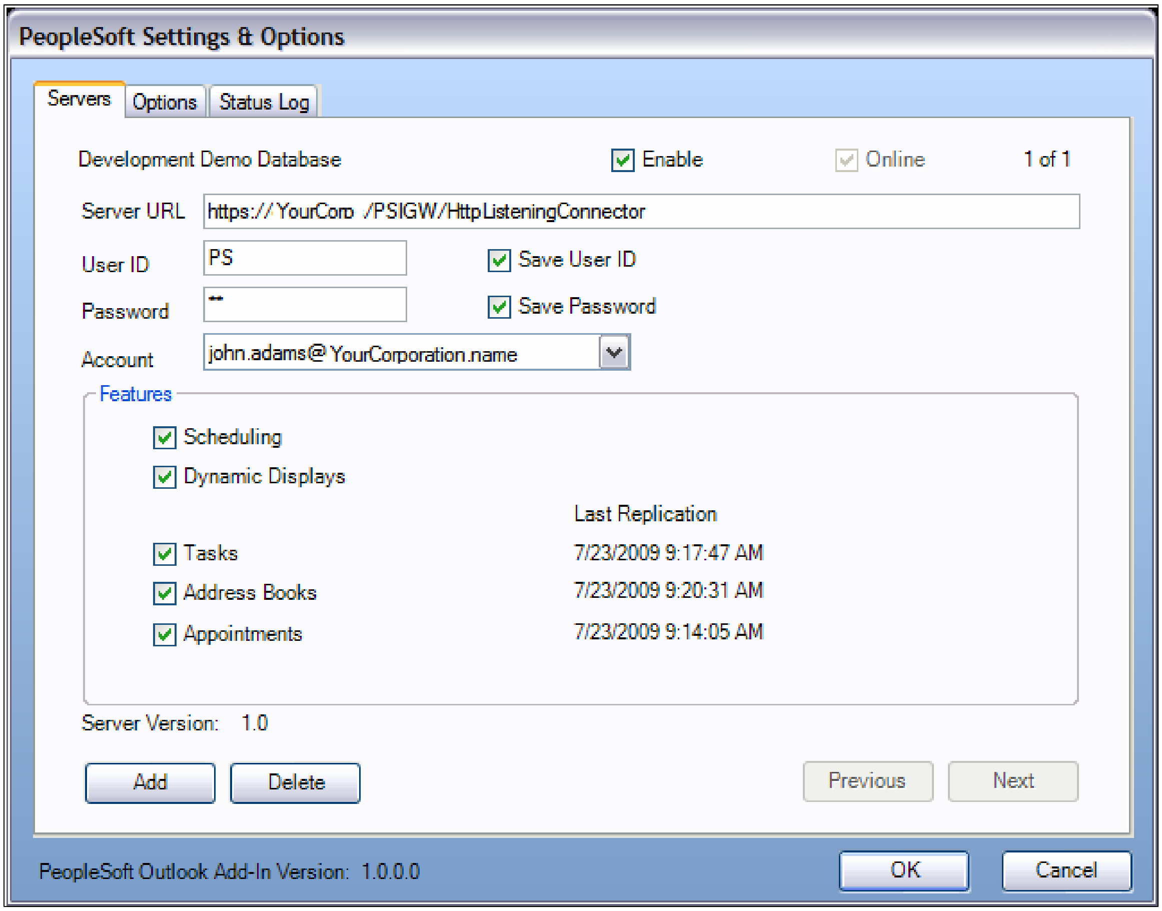
Defining the Servers Page

Access the PeopleSoft Settings and Options - Servers page (PeopleSoft Outlook menu, Settings & Options, click the Servers page).

Use the PeopleSoft Settings and Options - Servers page to define the server URL that you want to connect to and the functions that you want to assign to the server.

Image: PeopleSoft Settings & Options - Servers page

This example illustrates the fields and controls on the PeopleSoft Settings &amp; Options - Servers page. You can find definitions for the fields and controls later on this page.



Development Demo Database ☑ Enable ☑ Online 1 of 1

Server URL: https:// YourCorp /PSIGW/HttpListeningConnector

User ID: PS ☑ Save User ID

Password: ** ☑ Save Password

Account: john.adams@ YourCorporation.name

☑ Scheduling

☑ Dynamic Displays

	Last Replication
☑ Tasks	7/23/2009 9:17:47 AM
☑ Address Books	7/23/2009 9:20:31 AM
☑ Appointments	7/23/2009 9:14:05 AM

Server Version: 1.0

PeopleSoft Outlook Add-In Version: 1.0.0.0

Define the URL and functions to assign to the server:

Enable

The server must show this check box selected to attempt connectivity.

If the server was determined to be online during the last connection attempt, the Online check box is selected.

Server URL

Enter the URL of the server that you want to connect to. This is your Integration Broker URL.

User ID

Enter your User ID.

Save User ID

Select to save your User ID in the system and avoid the need for future entry.

Password

Enter your password.

Save Password

Select to save your password in the system and avoid the need for future entry.

Account

Select an account in the form of the email address of the user. You can select from a list of multiple accounts.

The selected account drives the information from the parameters that you defined in the Outlook setup.

Features

Select the features that you want to use. Values are:

- Scheduling
- Dynamic Displays
- Tasks
- Address Books
- Appointments

Add

Click to add a new server option.

> **Note:** This section of the PeopleSoft Settings & Options - Servers page shows the current version of the PeopleSoft Outlook Add-In and the server that you are using.

Delete

Click to delete an existing server option.

Alternatively, you can disable a server by clearing the Enable check box.

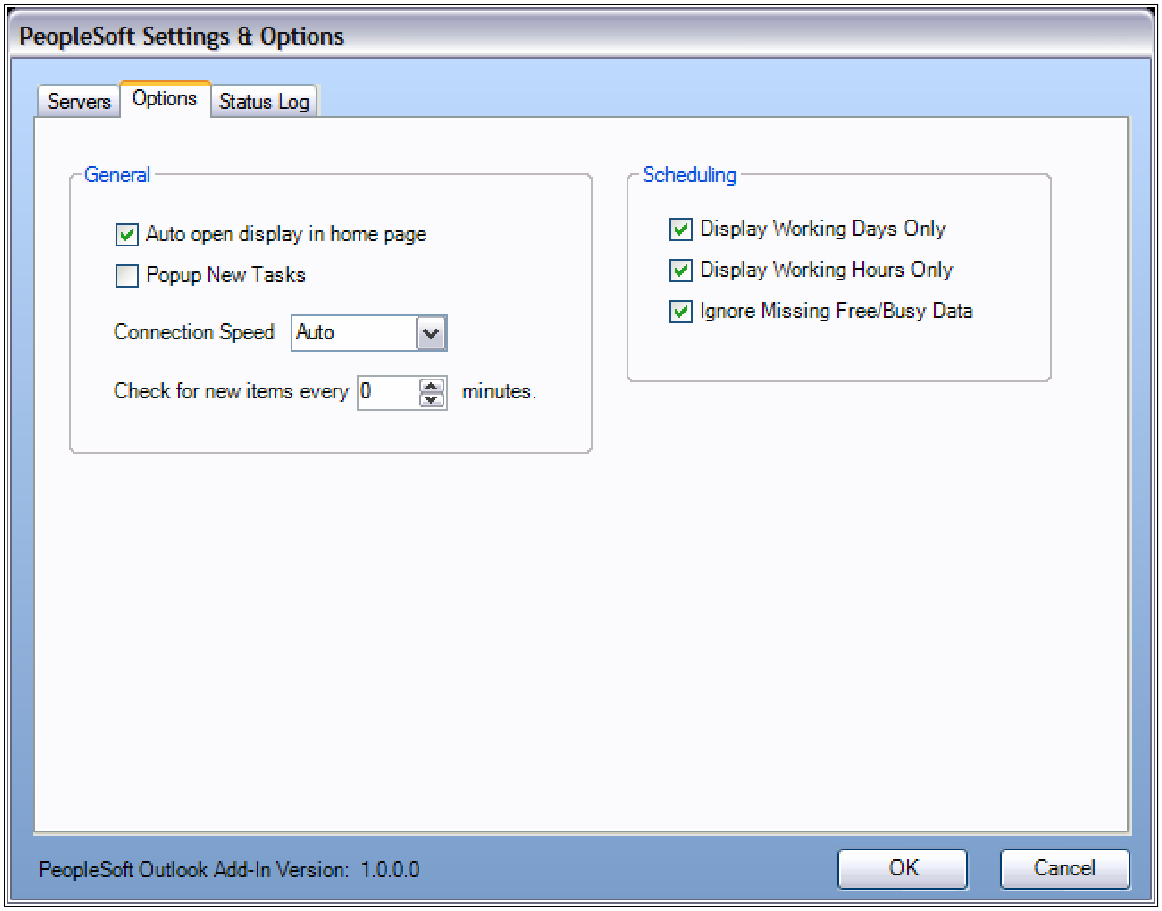
Defining the Options Page

Access the PeopleSoft Settings & Options - Options page (PeopleSoft Outlook menu, Settings & Options, click the Options page).

Use the PeopleSoft Settings & Options - Options page to review or select your display features and frequency of updates.

Image: PeopleSoft Settings & Options - Options page

This example illustrates the fields and controls on the PeopleSoft Settings &amp; Options - Options page. You can find definitions for the fields and controls later on this page.



General

☑ Auto open display in home page

□ Popup New Tasks

Connection Speed: Auto

Check for new items every 0 minutes.

Scheduling

☑ Display Working Days Only

☑ Display Working Hours Only

☑ Ignore Missing Free/Busy Data

PeopleSoft Outlook Add-In Version: 1.0.0.0

Auto open display in home page

Select to automatically open the display in your Microsoft Outlook home page.

Popup New Tasks

Select to be alerted by popup messages when a new task falls into your queue.

Connection Speed

Select to specify the connection speed. Values are:

- *Auto*: This is the default value.
- *Fast*
- *Average*
- *Slow*

You can adjust your connection speed which will adjust the chunk size when sending and receiving files.

In Desktop Integration's File Transfer page, an option exists to override your connection speed.

Check for new items every

Specify the frequency of system updates in minutes. This determines how often you want to replicate the PeopleSoft system information.

If set to *0,* no replication will take place.

Display Working Days Only

Select to only display working days by default in the PeopleSoft Scheduling Assistant.

Display Working Hours Only

Select to only display working hours by default in the PeopleSoft Scheduling Assistant.

Ignore Missing Free/Busy Data

Select for the system to ignore missing free/busy data.

When you select this option you avoid unnecessary clutter from unavailable data.

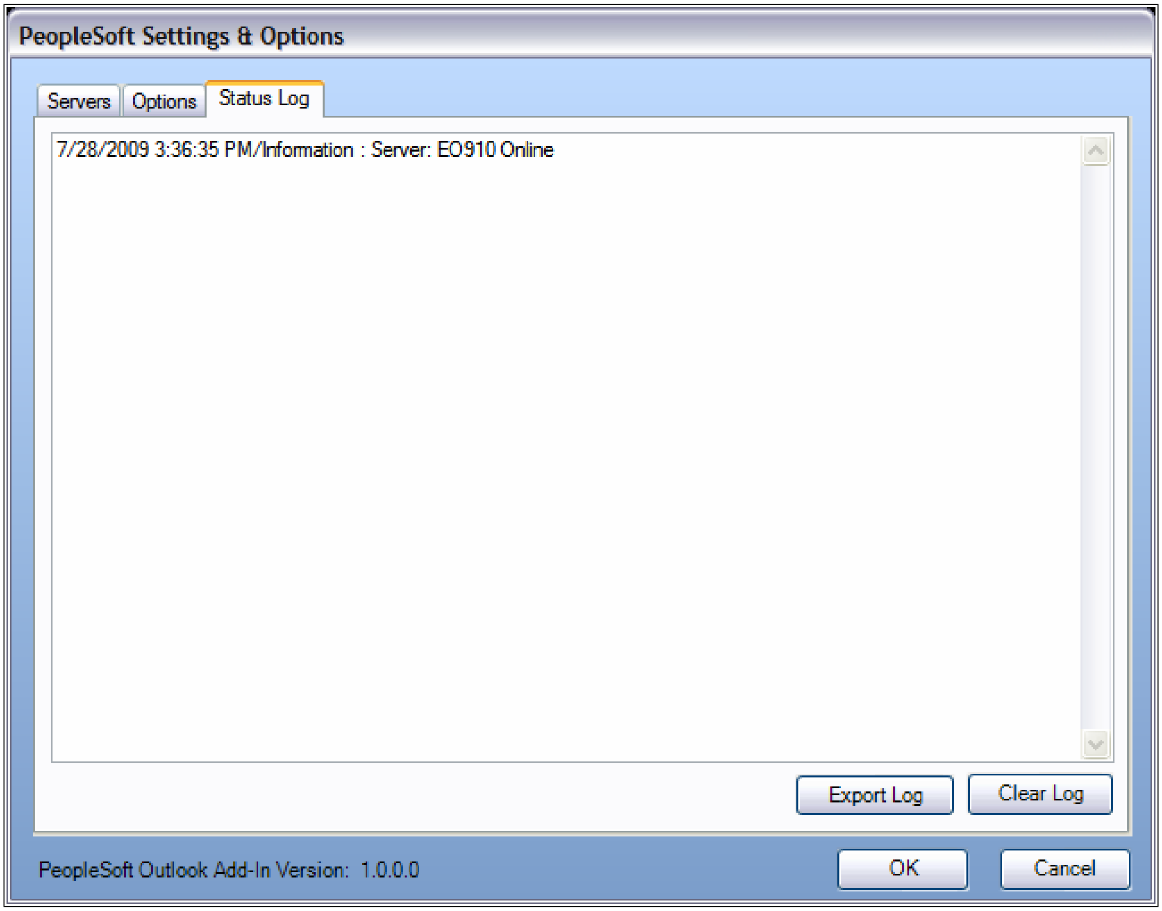
Reviewing the Status Log Page

Access the PeopleSoft Settings & Options - Status Log page (PeopleSoft Outlook menu, Settings & Options, click the Status Log page).

Use the PeopleSoft Settings & Options - Status Log page to review the send and receive data and any errors in the log.

Image: PeopleSoft Settings & Options - Status Log page

This example illustrates the fields and controls on the PeopleSoft Settings &amp; Options - Status Log page. You can find definitions for the fields and controls later on this page.



7/28/2009 3:36:35 PM/Information : Server: EO910 Online

PeopleSoft Outlook Add-In Version: 1.0.0.0

If you are experiencing any issues, you can access this page and clear the log or export and save the log information. This log is only available in the current session. If you exit Microsoft Outlook, and then log back in, the log information is erased and no longer visible. This page also displays the current version of the PeopleSoft Outlook Add-In that you are using.

Using the PeopleSoft Scheduler and Scheduling Assistant

Use the PeopleSoft Scheduler page and Scheduling Assistant to expand the functionality of your appointments, meetings, address books, and messages.

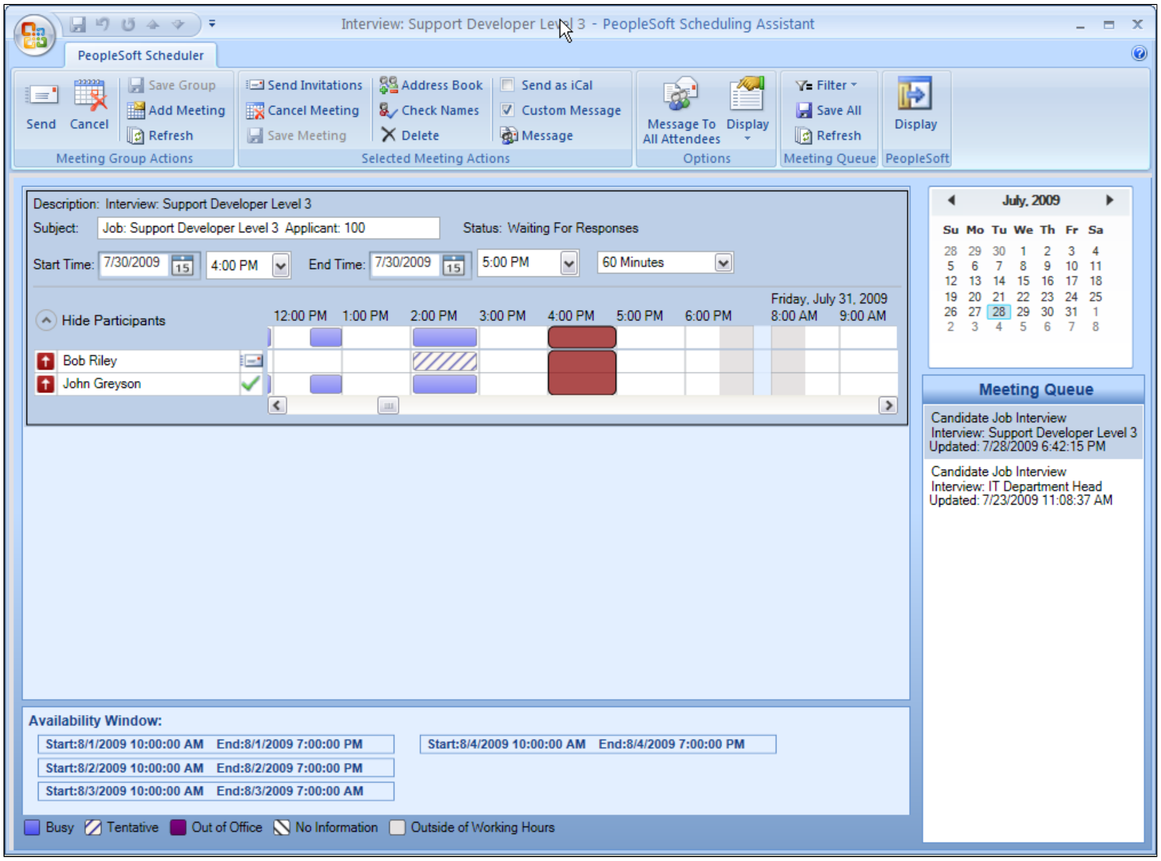
The PeopleSoft Scheduling Assistant helps you manage and schedule all of your PeopleSoft meetings.

Navigation

PeopleSoft Outlook menu, Scheduling Assistant

Image: PeopleSoft Scheduler page

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



Description: Interview: Support Developer Level 3

Subject: Job: Support Developer Level 3 Applicant: 100

Status: Waiting For Responses

Start Time: 7/30/2009 4:00 PM End Time: 7/30/2009 5:00 PM 60 Minutes

Hide Participants

Bob Riley

John Greyson ✓

Meeting Queue

Candidate Job Interview
Interview: Support Developer Level 3
Updated: 7/28/2009 6:42:15 PM

Candidate Job Interview
Interview: IT Department Head
Updated: 7/23/2009 11:08:37 AM

Availability Window:

Start:8/1/2009 10:00:00 AM End:8/1/2009 7:00:00 PM

Start:8/2/2009 10:00:00 AM End:8/2/2009 7:00:00 PM

Start:8/3/2009 10:00:00 AM End:8/3/2009 7:00:00 PM

Start:8/4/2009 10:00:00 AM End:8/4/2009 7:00:00 PM

🟦 Busy ▨ Tentative 🟪 Out of Office ▧ No Information □ Outside of Working Hours

The PeopleSoft Scheduler page and Scheduling Assistant provide the following features and functionality:

Meetings Group Actions

Send	Click to send the selected meeting or group of meetings.
Cancel	Click to cancel the selected meeting or group of meetings.
Save Group	Click to save the selected meeting or group of meetings.
Add Meeting	Click to add a meeting.
Refresh	Click to refresh. This action retrieves the data from the PeopleSoft system. If you have unsaved data the system prompts you to save now, or your data will be lost.

Selected Meeting Actions

Send Invitations

Click to send the selected invitation or group of invitations.

Cancel Meeting

Click to cancel the selected meeting or group of meetings.

Save Meeting

Click to save the selected meeting or group of meetings.

Address Book

Click to add someone from your address book to the selected meeting or group of meetings.

Check Names

Click to initiate the system to verify that your selected participants have valid email addresses.

Delete

Click to delete the selected meeting or group of meetings.

Send as iCal

Click to send the selected meeting as an iCalendar attachment.

Custom Message

Click to send a custom message, and then select Message to choose which message you want to use.

Message

Click to choose the message you want to use.

Options

Message To All Attendees

Click to send the same message to multiple people.

Display

From the drop-down list, select the features that you want to display in your Outlook homepage. Values are:

- Meeting Queue
- Availability Window
- Show Working Days Only
- Show Working Hours Only

Meeting Queue

When you look in the Meeting Queue display, you see everything that is assigned to you to be scheduled. You can click a specific meeting (line item) to display the meeting details in your Outlook homepage.

These options enable you to filter what is visible in your Meeting Queue.

Filter

Select from the available options to display in your Outlook homepage:

- *Ready to Schedule*: This is the first stage; in the Meeting Queue but nothing has been done.
- *In Progress*: Select to display meetings that are in progress. Invitations may have been sent, but these meetings are incomplete.

- *Scheduled Meetings*: Select to display meetings that have been scheduled.
- *Past Meetings*: Select to display meetings that already occurred.
- *Cancelled Meetings*: Select to display meetings that were cancelled.
- *On Hold Meetings*: Select to display meetings that were in the Meeting Queue, then put on hold.
- *Set Date Range*: Select to specify a date range for the options you want to display.

Save All Click to save everything selected in your Meeting Queue.

Refresh Click to refresh your Meeting Queue.

PeopleSoft

Display Displays Dynamic Displays for the current document.

Replicating Appointments, Meetings, Tasks and Address Books

The replication options in the PeopleSoft Outlook menu have no dialog or visual component. These items are replicated to and from PeopleSoft only. Therefore, only appointments, meetings, tasks, and address books that are also PeopleSoft items can be replicated. The menu options enable you to perform the replications manually.

Alternatively, you can set up a schedule for replication in the PeopleSoft Settings & Options pages.

See Defining Settings and Options.

Replicating Appointments and Meetings

Use the PeopleSoft Replicate Appointments/Meetings option to manually start the replication process for those items.

Replicating Tasks

Use the PeopleSoft Replicate Tasks option to manually start the replication process for those items.

Replicating Address Books

Use the PeopleSoft Replicate Address Books option to manually start the replication process for those items.

Chapter 11

Understanding Enterprise Integration

Understanding PeopleSoft Business Interlinks

PeopleSoft Business Interlinks enables you to perform component-based, real-time integration from PeopleSoft applications to external systems. PeopleSoft Business Interlinks creates synchronous transactions that enable PeopleSoft applications to pass data to and receive data from the external system in real time. You can use PeopleSoft Business Interlinks to integrate PeopleSoft with third-party systems, with another PeopleSoft application, or with systems on the internet.

Understanding PeopleSoft Component Interfaces

A component interface is a set of application programming interfaces (APIs) that you can use to access and modify PeopleSoft database information using a program instead of the PeopleSoft client. PeopleSoft Component Interfaces exposes a PeopleSoft component (a set of pages grouped for a business purpose) for synchronous access from another application (PeopleCode, Java, C/C++, or Component Object Model [COM]). A PeopleCode program or an external program (Java, C/C++, or COM) can view, enter, manipulate, and access PeopleSoft component data, business logic, and functionality.

Understanding File Layouts

A file layout is a definition (or mapping) of a file to be processed. It identifies where fields are located in file data. Once you create a file layout, you can write PeopleCode programs that ultimately use the file layout, either to read data from or write data to a file.

In addition to manipulating transaction data, you can use file layouts and flat files to move data between your PeopleSoft database and external systems (data interchange).

Understanding PeopleSoft Integration Broker

PeopleSoft Integration Broker is a messaging system that enables you to synchronize data in one application or system with another.

PeopleSoft Integration Broker facilitates synchronous and asynchronous messaging among internal systems and trading partners, while managing message structure, message format, and transport disparities.

PeopleSoft Integration Broker comprises two high-level subsystems: the integration engine and the integration gateway. The integration engine runs on the PeopleSoft application server. It is tied closely to PeopleSoft applications and produces or consumes messages for these applications.

The integration gateway manages the receipt and delivery of messages passed among systems through PeopleSoft Integration Broker. It provides support for the leading TCP/IP protocols used in the marketplace today, and more importantly, provides extensible interfaces for the development of new connectors for communication with legacy and internet-based systems.

Chapter 12

Understanding Integration Points

Overview of Integration Points

Provided by a PeopleSoft application, an integration point is an interface that is used to communicate with other PeopleSoft or external applications. An integration point provides integration details for PeopleSoft applications, including which technologies are involved, technology details, information for using PeopleSoft Integration Broker for messaging, and how different integration points are related.

The integration point consists of data rules for the applications that it supports. The integration points that are delivered with PeopleSoft applications provide generic functionality so that they can be adapted for use with as many programs as possible.

You can implement an integration point can be implemented by using different technologies available in PeopleTools, such as messaging, component interfaces, business interlinks, XML links, and electronic data interchange (EDI).

Integration points can be associated with or used by application groups. An application group is a logical grouping of applications that use an integration point in the same business manner.

Other than this grouping facility, an application group and an application mean the same thing. In the rest of the documentation, the words *application group* and *application* are used interchangeably unless clearly specified.

Every integration point is owned by at least one application, but can be used in multiple applications. Therefore, if an application sends an integration point, and another application can receive the same integration point, the two systems should be interoperable, assuming the data structure and the rules of the integration point are implemented the same in both places.

However, sometimes two applications might use the same integration point but implement it in different ways. For example, one application that uses the Customer integration point may need to transform the data before it can be sent to an external system, which has another data structure for its customer information.

An integration point can be a part of multiple application groups. For example, the Department Table integration point may be used by a number of application groups, including PeopleSoft Human Resources, Customer Relationship Management (CRM), and General Ledger.

For more information, refer to the Interactive Services Repository on the My Oracle Support website.

Chapter 13

Activating Integration Points

Setting Up PeopleSoft Integration Points

This section discusses how to:

- Set up publication rules.
- (Optional) Map nodes to a chunking rule.
- (Optional) Assign business units or setIDs to a chunking rule.
- Specify OnRoute PeopleCode.

Note: For more information and technical details about these integration points, and for information about how and with what applications to use them, consult the relevant application documentation.

Pages Used to Set Up PeopleSoft Integration Points

Page Name	***Definition Name***	***Navigation***	***Usage***
Batch Publish Rules	EOIU_SOPUBATCH	Enterprise Components, Integration Definitions, Batch Publish Rules	Set up publication rules. You must activate a publication rule for the publication messages that you create to follow. This rule includes instructions on message chunking.
Add Nodes to Chunk Rule	EOIU_ADNODECHUNK_PNL	Enterprise Components, Integration Definitions, Map Chunking Rules, Node to ChunkRule	Map PeopleSoft message nodes to chunking rules.
Batch Publish	EOIU_BATCHPUB	Enterprise Components, Integration Definitions, Initiate Processes, Batch Publish	Create the run control for the batch publish process.

Batch Publish Rules Page

Use the Batch Publish Rules page (EOIU_SOPUBATCH) to set up publication rules.

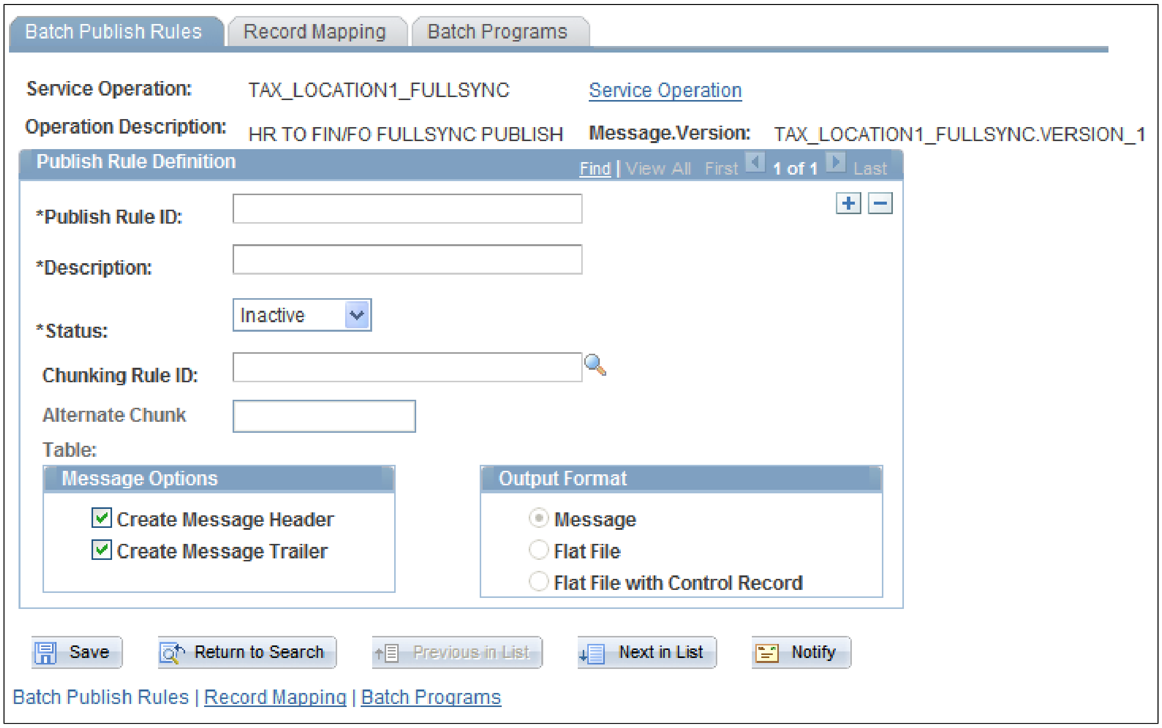
You must activate a publication rule for the publication messages that you create to follow. This rule includes instructions on message chunking.

Navigation

Enterprise Components, Integration Definitions, Batch Publish Rules

Image: Batch Publish Rules page

This example illustrates the fields and controls on the Batch Publish Rules page. You can find definitions for the fields and controls later on this page.



Service Operation: TAX_LOCATION1_FULLSYNC

Operation Description: HR TO FIN/FO FULLSYNC PUBLISH

Message.Version: TAX_LOCATION1_FULLSYNC.VERSION_1

Publish Rule Definition

*Publish Rule ID:

*Description:

*Status: Inactive

Chunking Rule ID:

Alternate Chunk Table:

Message Options

☑ Create Message Header
☑ Create Message Trailer

Output Format

● Message
○ Flat File
○ Flat File with Control Record

If the data that you're transmitting does not fit in a single message, or if you want to send different parts of the message to different target systems, set up the rules to chunk the message and associate it with the publish rule. The business unit and setID chunking rules are standard in PeopleSoft applications, but you can configure chunking rules.

Publish Rule ID	Select the name of the message for which you're setting up rules.
Status	Select *Active* to activate this publish rule definition for this message. Select *Inactive* to prevent this rule from applying to this message.
Chunking Rule ID and Alternate Chunk Table	Enter the unique chunking rule name that is set up when you created the chunking rule. The message that you publish is routed based on this field. If necessary, enter an additional field in the Alternate Chunk Rule ID field by which to chunk the message.

Message Options

Many PeopleSoft systems rely on a message header and message trailer to trigger subscription PeopleCode to discard old table data and insert the new incoming data. As a general rule, all FullSync messages should use a header and trailer. Sync messages don't need headers and trailers.

Output Format

The Application Engine program used to chunk messages can create either an XML message that flows through messaging architecture or a flat file that is generated in PeopleSoft Process Scheduler and not published elsewhere. Always select Message as the format when you send data to PeopleSoft systems.

Add Nodes to Chunk Rule Page

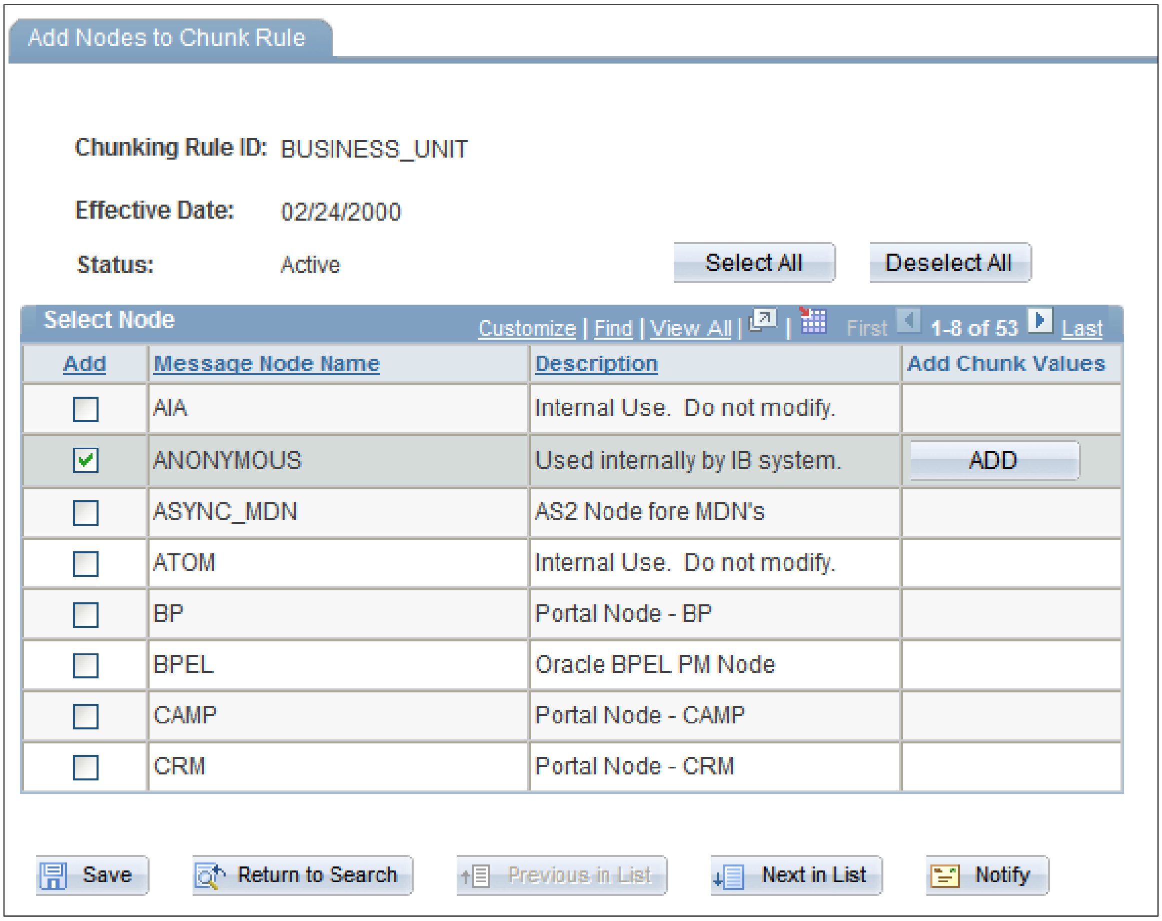
Use the Add Nodes to Chunk Rule page (EOIU_ADNODECHUNK_PNL) to map PeopleSoft message nodes to chunking rules.

Navigation

Enterprise Components, Integration Definitions, Map Chunking Rules, Node to ChunkRule

Image: Add Nodes to Chunk Rule page

This example illustrates the fields and controls on the Add Nodes to Chunk Rule page. You can find definitions for the fields and controls later on this page.



Chunking Rule ID: BUSINESS_UNIT

Effective Date: 02/24/2000

Status: Active

Add	Message Node Name	Description	Add Chunk Values
□	AIA	Internal Use. Do not modify.	
☑	ANONYMOUS	Used internally by IB system.	ADD
□	ASYNC_MDN	AS2 Node fore MDN's	
□	ATOM	Internal Use. Do not modify.	
□	BP	Portal Node - BP	
□	BPEL	Oracle BPEL PM Node	
□	CAMP	Portal Node - CAMP	
□	CRM	Portal Node - CRM	

To map nodes to a chunk rule:

1. In the Add column, select the check box next to the nodes that you defined earlier.

   After you select a node, use the Add button in the Add Chunk Values column to open the Quick Map page for the message you defined earlier.

2. Click Save.

Assigning Business Units or SetIDs to a Chunking Rule

See Assigning Business Units or SetIDs to a Chunking Rule.

Specifying OnRoute PeopleCode

Create a service operation handler to route the message chunk to the correct subscriber node.

To do so, extend the IRouter application class and use the OnRouteSend method. Then in the appropriate service operation, define a handler and specify the application class.

Setting Up Related Languages

This section provides overviews of related language tables and related language guidelines for PeopleSoft messaging and discusses how to:

- Interpret component processor behavior.
- Publish a message from a component.
- Publish a message from batch programs.
- Subscribe to data in a PeopleSoft multilingual environment.
- Subscribe to data in a non-multilingual environment.

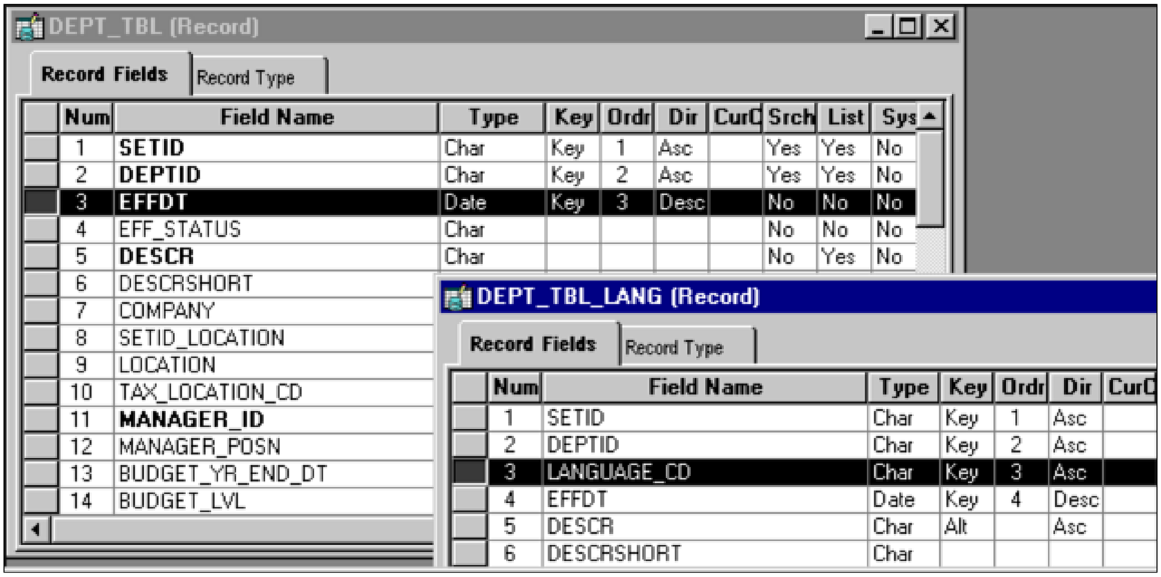
Understanding Related Language Tables

You can use several possible scenarios to familiarize yourself with related languages when setting up related languages for a message.

A department table, for example, must publish information in German as well as English. In the following example, the base application tables and related-language tables have a parent-child relationship. The related-language table has the same name as the parent table, but is suffixed with _LANG, in accordance with PeopleSoft naming conventions.

Image: Related-language record definitions

This example illustrates the fields and controls on the Related-language record definitions. You can find definitions for the fields and controls later on this page.



Num	Field Name	Type	Key	Ordr	Dir	CurC	Srch	List	Sys
1	**SETID**	Char	Key	1	Asc		Yes	Yes	No
2	**DEPTID**	Char	Key	2	Asc		Yes	Yes	No
3	**EFFDT**	Date	Key	3	Desc		No	No	No
4	EFF_STATUS	Char					No	No	No
5	**DESCR**	Char					No	Yes	No
6	DESCRSHORT								
7	COMPANY								
8	SETID_LOCATION								
9	LOCATION								
10	TAX_LOCATION_CD								
11	**MANAGER_ID**								
12	MANAGER_POSN								
13	BUDGET_YR_END_DT								
14	BUDGET_LVL								

Num	Field Name	Type	Key	Ordr	Dir	CurC
1	SETID	Char	Key	1	Asc	
2	DEPTID	Char	Key	2	Asc	
3	LANGUAGE_CD	Char	Key	3	Asc	
4	EFFDT	Date	Key	4	Desc	
5	DESCR	Char	Alt		Asc	
6	DESCRSHORT	Char				

Consider the following points:

- The parent table, DEPT_TBL, has the related-language child table, DEPT_TBL_LANG.
- The child table has the same fields as the parent table, plus an additional field of LANGUAGE_CD.
- The attributes of the child table are all of the translatable textual fields of the parent record.

Understanding Related Language Guidelines for Messaging

When publishing a full message, generate messages that contain the contents of an entire table by first generating a message in the base language of the system that contains the full table contents. Then generate messages for each of the related languages that are supported by the publishing system. Each message should contain the full message structure for that message object (levels 0, 1, 2, and so on). The language-specific messages should contain the translatable field values for that language and include the base language fields that are not translatable.

When subscribing to a full message, specify the language code only at level 0 of the message. This captures and sets the user’s preferred language to level 0 of the PeopleSoft Common Application Message Attributes (PSCAMA) message header. All data within the message must be in the same language. Follow these steps:

1. Delete the base language tables and related-language tables.
2. Replace these tables with data from the messages as appropriate.
3. Place only those related-language field values that are supported by the subscribing system into the related-language tables.

4. Add the related-language table entry only if the base language table entry already exists.

When publishing an incremental message, the PeopleSoft system generates base messages in the user's preferred language by using the language code of the user ID. Putting the user’s preferred language code in the message header PSCAMA record defines the message language for the subscribing system.

When subscribing to an incremental message using PeopleSoft Component Interfaces, use a simple PeopleCode program that performs a SetLanguage (messagelanguage) call to a component interface with the message definition. This enables the subscribing system to process the data in the appropriate related or base language for that system.

When subscribing to an incremental message using PeopleCode only, the PeopleCode program must simulate what the component processor does. The PeopleSoft system provides a generic Subscribe_IncrReplication PeopleCode function that provides basic language-related ability for incremental message subscriptions.

Note: All PeopleSoft subscription processes that are associated with textual information work as if the content is related-language enabled; thus the processes provide support for customer related-language extensions and future PeopleSoft enhancements.
For PeopleSoft-to-PeopleSoft system integration, you do not need to specify the language-sensitive data on either system.

All of the PeopleCode functions that are needed for related language processing of incremental and full messages are in the FUNCLIB_EOEIP record. The record contains two functions:

- Subscribe_IncrReplication has related-language processing for an incremental message subscription process.
- Subscribe_FullReplication has related-language processing for a batch subscribe process.

Interpreting Component Processor Behavior

When you open a component, the component processor:

1. Compares the user's preferred language against the base language for the database.
2. Uses the record information from the base application table (DEPARTMENT_TBL).

   If a record in the base application table exists for the user's preferred language, the fields on the related-language table (DEPARTMENT_LANG) overlay the record information. For example, a German user sees German descriptions even if the base language for the database is English.

When you change the user's preferred language and save the component:

1. The component processor writes all the data for related-language fields back to the related-language table.
2. The component processor writes the rest of the data back to the base application table.
3. The German user’s entries for the DESCR and DESCRSHORT fields are saved back to the DEPARTMENT_LANG table with its key values and the LANGUAGE_CD field in German.

4. The data that was entered by the German user in the key fields, as well as MANAGER_NAME and ACCOUNTING_OWNER fields, are saved on the parent record DEPARTMENT_TBL.

Publishing a Message from a Component

The PeopleSoft system employs the user's preferred language to determine the language of a message that is published from a component. The default for LANGUAGE_CD is set to the preferred language code (OPERATOR.LANGUAGE_CD). The standard for incremental changes is to publish only the data that has changed, in the language to which it was changed. Changing the preferred language to translate data generates new messages appropriately.

Publishing a Message from Batch Programs

Application Engine and Structured Query Report incremental message programs should use the base language of the system. These programs perform their accesses and updates on the base tables only, even if related-language tables are supported for those business objects. Related-language tables are featured in batch program processing only in generating warnings or errors that use the message catalog.

When a batch application program runs, the processing is done in the base language of the system, and messages are generated in only the base language.

Subscribing to Data in a PeopleSoft Multilingual Environment

The subscription process sets the language for processing the message to the language needed by the subscriber system. For example, if the base language of the subscribing system is French, and the PeopleSoft system is sending German data, the subscription process must store the German data in a related-language table and the nontranslatable data in the French base application tables. Neither system's base language matters; only the base language of the subscribing system and the actual message language are used.

Subscribing to Data in a Non-Multilingual Environment

When handling subscribing systems that do not support multiple languages, you can subscribe to data in these ways:

- Setting message publish routing PeopleCode.
- Sending service operations to all appropriate nodes.
- Permitting the subscription to process itself.
- Subscribing to data that is specific to an external system.

Setting Message Publish Routing PeopleCode

Set message publish routing PeopleCode to send only messages in a particular language code to a subscribing node.

The subscribing node does not need to check the language in which the message was generated; any message that it receives in its language is automatically used to update the subscribing system’s database. To implement this option:

- Determine the languages in which messages are published.
- Determine the message language that a subscribing node receives.
- Add PeopleCode routing logic on the publish side to check the language code of the first occurrence in the message record and return a list of subscribing nodes that should receive the message for that language.

The publish routing PeopleCode guarantees that the message is sent to the correct subscribing nodes by using the message language code.

Sending Service Operations to All Appropriate Nodes

Send service operations to all appropriate nodes regardless of the language and have the subscription routing PeopleCode filter out messages in different languages.

PeopleCode on the subscription routing checks the language code of the first occurrence in the message and controls whether the node should receive the message. To implement this option:

- Ensure that service operations in all languages are sent to all appropriate nodes.
- Add the PeopleCode to compare the message language against a hard-coded language value for the subscribing system.

The advantage of putting the logic within the subscription routing PeopleCode is that every message is checked for a language value match.

Permitting the Subscription to Process Itself

You can permit the subscription process itself, rather than the routing PeopleCode for the service operation and queue, to determine whether the message should be processed for the subscribing system.

The subscription process checks the language code of the PSCAMA record for the first instance of the message against a hard-coded value for the subscribing system. If the language code does not match, the message is ignored. If the message language code does match, it's considered a base language message, and it replaces all data on the subscribing system according to the audit action flags on the message records.

Note: To prevent data integrity issues, generic subscription processes should not filter messages based on language code.

Subscribing to Data That Is Specific to an External System

Subscribe to data that is specific to an external system for language code, business unit, or setID requirements that are specific to an external system.

Use the chunking rule and the routing control tables that Oracle supplies to select a portion of the data and send it to a specific node.

Use the PeopleSoft-supplied Publish Header (PublishHdr) component to enter the partitioning views and fields for a message. This chunks the message so that all contents within a single message contain the same partitioning value (such as business unit, setID, or application-specific fields).

Set up the routing control for the message so that the message is sent only to the appropriate nodes. The PeopleSoft system supplies a business unit routing control (BU Routing Control) component and a setID routing control component that enable applications to specify for each message which nodes should receive the partitioned message data.

After you set up the chunking rule and routing rules, both the full data publish and batch publish programs partition the data according to the appropriate value and route it accordingly. You can now publish and subscribe to the message.

Examining Related-Language Messaging Scenarios

Actual related-language messaging scenarios include publishing a non-base language message and switching a preferred language.

This section discusses how to:

- Publish a non-base language message to a PeopleSoft subscribing system with a different base language and no prior data.
- Switch the preferred language to Japanese and update the same employee's name and address.

Publishing a Non-Base Language Message to a Subscribing System with a Different Base Language and No Prior Data

In the following example, the publishing system base language is English, the subscriber base language is Japanese, and the user’s preferred language is German.

This is the process for publishing:

1. An online user adds a new level 0 key on a page.

   The data is stored in both the base language table (English) and the related-language table (German).

2. The system publishes a message in the user’s preferred language (German).
3. The data is inserted into the subscribing system.

   The data is inserted into both the base language table and related-language table (German) because it is added for the first time; data cannot reside in a related-language table without corresponding data being in the base table.

Switching the Preferred Language to Japanese and Updating the Same Employee's Name and Address

This is the process for switching a preferred language:

1. A user switches the preferred language from English to Japanese.

   The user updates the same record as in the previous scenario. Data that is not language-sensitive is updated in the base table (English). Language-sensitive data is inserted into the related-language table (Japanese).

2. A system publishes a message in the user’s preferred language (Japanese).

3. Data is inserted into the subscribing system.

All data goes to the base table (Japanese), because the message was sent in the same language as the subscribing system’s base language.

Appendix A

Integration Point Naming Standards

Standard Action and Event Verbs

The following standard verbs are used in the names and descriptions of integration points:

Verb	***Description***
Acknowledge	Indicates receipt of a processing request. Also conveys the result of the original request. (For example, your application acknowledges a purchase order (PO) when a PO has been issued and the corresponding business application acknowledges the receipt of the PO and responds with an acceptance or counter offer.)
Add	Use when a complete entity has already been constructed and needs to be communicated for the first time to another business application, and when business implications go beyond what a Sync message would convey. Transactional messages may use Add; setup data should use Sync.
Adjust	Clarifies a specific process (for example, adjustment of inventory quantity on hand when neither Add, Change, or Delete conveys the full meaning).
Allocate	Clarifies a specific process (for example, allocating costs to different business applications when neither Add, Change, or Delete conveys the full meaning).
Approve	Use when an entity passes an approval process and is ready for the next business process step.
Cancel	Use when business implications go beyond a simple change or delete (for example, canceling a purchase order).
Change	Use when an entity is changed based on a business event and the change needs to be communicated to another business application. Encompasses business implications beyond what a Sync message would convey (for example, SalesOrderChange). Transactional messages can use Change; setup data should use Sync.
Confirm	Responds to a request from the receiving application to confirm. This function conveys the result of the original request (for example, when an inventory issue must be confirmed in an application based on an event in a warehouse management business application).

Verb	***Description***
Create	Use when the processing must initiate the building of the document rather than moving the document from one system to another.
Find	Requests a list of items from a business application. The response to this request is List. Equivalent to a search dialog box in which you pass the search criteria over as the message.
FullSync	Replicates a complete entity, including all record instances, between business applications to initially seed the receiving application with that data. Use with all full message definitions.
Get	Requests a specific data entity from a business application. The response to this request is Show. Differs from Find in that the specific entity's key values are known and its details are being requested, whereas Find checks for existence and returns a list of values that match the Find criteria.
Issue	Clarifies a specific process (for example, issue material from inventory in cases in which neither Add, Change, or Delete conveys the full meaning).
List	Use when sending a list of multiple data entities in a summary format. The List verb can be used to respond to a Find or Get request, or in a publish scenario, to push information to other applications based on a business event. The results of a List can be used as is, or they can be used to select a specific instance of a document or entity to issue a detail Get request.
Load	Initiates the addition of a data entity to another business application where maintenance of the document passes to the receiving application permanently. When the request is passed, the sending application no longer has direct control over the document or entity.
Post	Clarifies a specific process (for example, a posted journal entry, in cases in which neither Add, Change, or Delete conveys the full meaning).
Receive	Clarifies a specific process (for example, when you receive inventory against a PO, in cases in which Change is not detailed enough for the business context).
Request	Requests specific data from a business application. The requested data should be passed back as a separate message.
Show	Sends information about a specific instance of a business entity. Can also be used to respond to a Get request, or in a publish scenario in which it pushes information to other applications based on a business event.

Verb	***Description***
Sync	Communicates the need to update master files between business applications. Facilitates application integrity and ease of data entry for the business user by enabling a single point of input. Should contain only incremental messages of Add, Change, and Delete actions to the entity. Normally used for passing setup messages between applications.
Transfer	Clarifies a specific process (for example, in the event of a transfer of material from one inventory location to another, in cases in which neither Add, Change, or Delete conveys the full meaning).
Update	Clarifies a specific process (for example, in the event of an update of inspection information from one business application to another, in cases in which neither Add, Change, or Delete conveys the full meaning). The event is not adding a document or changing fields, but communicating the occurrence of an event as well as the corresponding data that accompanies the event. Transactional data may use Update; setup data should use Sync.

Standard Business Object Nouns

Business objects appear in uppercase with underscores between key words.

The following table lists examples of business object names only; it is not intended to list or set conventions.

Account_Chartfield	***Deal***	***Names_Prefix_Suffix***	***Project_Category***
Action_Reason	Department	Nations_Duevo	Project_Status
APE_Industry	Dept_Budget	Occupation_Illness	Project_Team
Applicant	Detail_Calendar	Par_Location	Project_Type_Cat_ Link
Bank	DirectDeposit	Par_Location_Count	Purchase_Order
BOM	Earnings	Payroll	Rating_Model
Budget	Expense	Payroll_Paysheet	Regulatory_Region
Bus_Unit_FS	Expense_Advance	Pension_Fund	Resource_Category
Bus_Unit_GL	Expense_Report	Person_ Accomplishment	Resource_SubCategory
Bus_Unit_HR	Expense_Sheet	Person_Competency	Resource_Type
Bus_Unit_PC	Fund	Person_Contract_ Belgium	Review_Scale
Bus_Unit_PF	Grant	Person_Credit_Card	Salary

Account_Chartfield	***Deal***	***Names_Prefix_Suffix***	***Project_Category***
Carrier	Image	Person_Disability	Salary_Matrix
Company	Industry_Inspection	Person_Diversity	Salary_Plan
Company_Credit_ Card	InterUnit	Person_Education	Salary_Structure
Company_Property	Inventory	Person_Names	SalesOrder
Competency	Item	Person_Property	Schedule
Consumer	Item_BusUnit	Person_PriorWork	State
Consumer_Usage	Item_Rev	Person_Visa_Citizen	State_Name
Contract	Item_Vendor	Position	Statute
ContractBelgium	Job_Code	Product_Chartfield	Statute_Belgium
Country	Journal	Product_Item	Vendor
Customer	Labor_Category	Product_Group	Visa_Permit
Credit_Card	Labor_Relations	Product_Price	Voucher
Credit_Card_ Merchant	Location	Product_UOM	Union
Currency	Market_Price	Project	UOM
Customer	Market_Rate	Project_Activity	Workforce

Appendix B

PeopleSoft Design Patterns

List of Design Patterns

The following table presents brief descriptions of the delivered design patterns:

Design Pattern Name	***Description***
AE Row By Row Publish (application engine row by row publish)	In this design pattern, the transaction or setup data that you want to send out of the PeopleSoft system is updated by use of an Application Engine program that performs procedural (row-by-row) processing; you want to publish these changes. Generally, messages are used with this design pattern.
Batch Publish	Use this design pattern to publish messages from a batch application. The batch application can be a COBOL or Structured Query Report program that takes either a procedural or set-based approach, or it can be an Application Engine set-based program.
Batch Subscribe	This design pattern enables you to perform edits against messages in sets. This can be a useful technique for high volume data, including millions of inbound rows. This design pattern is useful when you know that a single message definition may contain multiple instances of a transaction, or when you must reuse an existing batch program.
CI Subscribe (component interface subscribe)	This design pattern uses a component interface to edit incoming message data. This enables you to reuse existing business rules when processing data.
Component Publish	In this design pattern, the transaction or setup data that you want to send out of PeopleSoft is being updated by using a PeopleSoft component. In this case, the data is already in the component buffer, and the Publish PeopleCode function is used to publish a message.
EDI In	This design pattern is for inbound EDI documents. You should use EDI only for existing EDI manager inbound transactions that must be supplied to an EDI partner, and you want to allow subscription to an XML message, or when you need to comply with other industry standards, such as SWIFT, BAI, or HL7, that have an existing EDI manager inbound map, and you want to convert to subscribing to an XML message.

Design Pattern Name	***Description***
EDI Out	This design pattern is for outbound EDI documents. You should use EDI only for existing EDI manager inbound transactions that must be supplied to an EDI partner and you want to allow subscription to an XML message, or you need to comply with other industry standards, such as SWIFT, BAI, or HL7, that have an existing EDI manager inbound map, and you want to convert to publishing to an XML message.
Full Table Publish	Use this design pattern to populate an entire copy of a table onto a remote database or legacy system. Generally, full data replication occurs with setup tables, or relatively static, low-volume tables that are keyed by setID. When a copy of a table exists on the remote system, incremental updates can be used.
Full Table Subscribe	Use this design pattern to subscribe to messages that contain an entire copy of a table that is published from a remote database or legacy system. Generally, full data replication occurs with setup tables, or relatively static, low-volume tables that are keyed by setID. When a copy of a table exists on the remote system, incremental updates can be used.
No Pattern	No design pattern is specified.
PeopleCode Subscribe	Use this design pattern to subscribe to messages by using a PeopleCode program when additional processing is required. Use PeopleCode subscription when simple edits or no edits against the inbound data are needed before you insert the data into the application tables or staging tables.
Sync Reply	In this design pattern, another system initiates a request for information from the PeopleSoft system and waits for information to be returned. This information must be provided by the PeopleSoft system in a real-time synchronous mode and in a conversational style of interface before the other system can continue processing. Generally, business interlinks are used to satisfy this type of request.
Sync Request	Use this design pattern when a PeopleSoft application must call a third-party vendor's application to request information. This information must be provided in a real-time, synchronous mode. Generally, business interlinks are used to satisfy this type of request.
XML Reply	In this design pattern, another system initiates a request for information from the PeopleSoft system. This information must be provided by the PeopleSoft system in a real-time synchronous mode and in a conversational style of interface before the other system can continue processing. Generally, XMLDocs are used to satisfy this type of request.
XML Request	Use this design pattern when a PeopleSoft application must call a third party vendor's application to request information. This information must be provided in a real-time, synchronous mode. Generally, XMLDocs are used to satisfy this type of request.