ORACLE
PeopleSoft

PeopleTools 8.62: Documents Technology

December 2025

ORACLE

Documentation Accessibility

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

Access to Oracle Support

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

Contents

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

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

Hosted PeopleSoft Online Help

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

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

Locally Installed PeopleSoft Online Help

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

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

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

Downloadable PeopleBook PDF Files

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

Common Help Documentation

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

- Application Fundamentals

- Using PeopleSoft Applications

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

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

Field and Control Definitions

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

Typographical Conventions

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

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

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

ISO Country and Currency Codes

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

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

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

Region and Industry Identifiers

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

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

Region Identifiers

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

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

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

- USF (U.S. Federal)

- E&G (Education and Government)

Translations and Embedded Help

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

Using and Managing the PeopleSoft Online Help

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

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

PeopleTools Related Links

PeopleTools 8.62 Home Page

PeopleSoft Search and Insights Home Page

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

PeopleSoft Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

Contact Us

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

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

Follow Us

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

Chapter 1

Getting Started with PeopleSoft Documents Technology

PeopleSoft Documents Technology Overview

PeopleSoft documents technology includes a Document Builder, a PeopleCode API, and several utilities that enable you to create, manage, and test documents. The Document Builder enables you to build documents from the ground up, by importing schema definitions, or from PeopleSoft table definitions. A PeopleCode API is provided to enable you to populate and retrieve document data. PeopleSoft delivers a number of document utilities to help you validate documents during and after construction; validate schema; resolve copy project, import, and upgrade metadata issues; and more. You can use documents for integrating with third-party partners, as an alternative to using standalone rowsets, as a mechanism to distribute complex data, and more.

Implementing PeopleSoft Documents Technology

This section contains information to consider before you use the PeopleSoft documents technology.

Determining Security Requirements

A security analyst should evaluate your security requirements for your documents implementation. PeopleSoft Documents Technology enables you to limit write access to documents as well as contain them in specific document packages.

Assessing Staff Skills

Assess the skills of the people who will perform development and administrative functions. Developers should have familiarity, training, or experience in the following areas:

- PeopleTools.
- PeopleCode.

In addition, developers should understand and have research capabilities in Extensible Markup Language (XML).

Other Sources of Information

In addition to the considerations presented in this topic, take advantage of all PeopleSoft sources of information, including installation guides, release notes, product documentation, curriculum, and red papers.

Chapter 2

Understanding PeopleSoft Documents Technology

PeopleSoft Documents

A PeopleSoft document is a managed object. It consists of two components, a logical component and a physical component or representation. PeopleSoft provides a Document Builder for creating and managing documents. Document utilities are also provided to help you test documents, test document schemas, identify and resolve metadata issues resulting from copy project and upgrade processes, and more.

Document Components

A PeopleSoft document is made up of two components: a logical component and a physical component.

Logical Documents

The logical component of a document, or logical document, specifies the abstract structure of data. It does not specify any information about persistence, formatting, rendering, behavior, and so on.

When you work on the logical aspect of a document, you manage document elements types and data types, define element and data type properties, and so on.

Physical Documents

The physical component of a document, or physical document, specifies concrete details about a document, such as database tables, XML tags names, JSON tag names, and so on.

When you work with the physical aspect of a document, you manage schema details, such as target and imported namespace information, whitespace, blank element filters, and more.

Physical Document Formats

When building documents in the PeopleSoft system, you can create the following document formats:

- XML-formatted documents.
- Relational-formatted documents
- JSON-formatted documents.

When you build out the format of a document, you are working with the physical document.

XML-Formatted Documents

In an XML-formatted document, you manage XML schema details, such as target namespaces, whitespace, filtering blank elements, and so on. You also manage element tag names, prefixes, and more.

Relational-Formatted Documents

In a relational-formatted document, you map PeopleSoft records to documents.

To create a map, you map a document to a PeopleSoft record and then map the elements of that document to the fields of the PeopleSoft record. This mapping enables you to populate database rowsets using documents.

Creating relational maps for documents is optional.

JSON-Formatted Documents

In a JSON-formatted document you manage tag names.

The Document Builder

PeopleSoft provides a Document Builder that enables you to create and manage documents and define XML and relational formats for them.

Document Utilities

The following table lists and briefly describes document technology utilities that PeopleSoft delivers:

Utility	***Description***
Document Schema Tester	Test document schemas against XML.
Document Tester	Build out document contents.
Create Document from Schema	Create documents from existing XML schemas (XSD).
Create Document from Record	Create documents from PeopleSoft records.
Update Target Location	Locate document definitions with invalid target locations.
Document/Metadata Validation	Check the integrity of metadata references to documents.

Translating Document Builder and Document Utility Fields

Fields on the Document Builder and document utility pages are not translatable.

Chapter 3

Configuring the System for Managing Documents

Understanding Configuring the System for Managing Documents

Before you can use the PeopleSoft system to build and manage XML documents or JSON documents, you must set a target location for the system to use for building and validating XML document schemas. In addition, you must set a namespace for XML document schemas.

Setting target locations and setting namespaces is described in further detail elsewhere in the product documentation

Related Links

“Setting Target Locations for Services” (Integration Broker Administration)

“Setting Service and Schema Namespaces” (Integration Broker Administration)

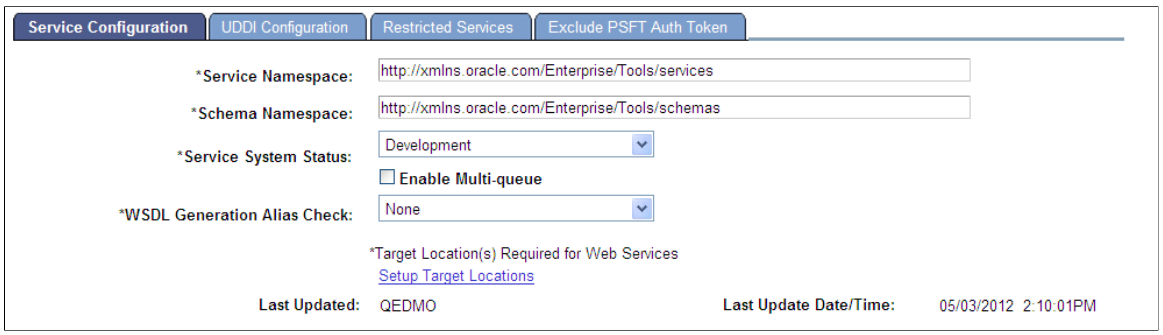
Setting Service and Schema Namespaces

To set the service and schema namespaces, use the Document Setup Options page (IB_DOCUMENT_OPTION). To access the page, select **PeopleTools** > **Documents** > **Documents Administration** > **Document Setup Options.**

This example illustrates the Document Setup Options page.

The Document Setup Options page features a **Service Configuration** link. When you click the link, the Service Configuration page (IB_SVCSETUP) appears.

This example illustrates the fields and controls on the Service Configuration page.



The steps for specifying service and schema namespaces are described elsewhere in the product documentation.

Related Links

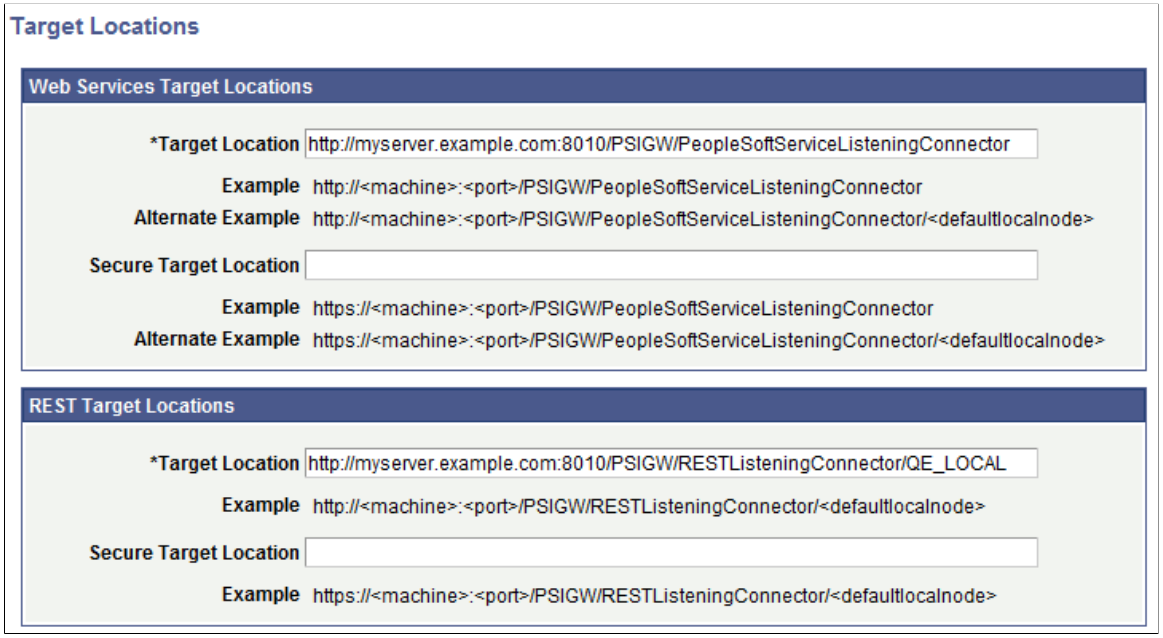
"Understanding Configuring PeopleSoft Integration Broker for Handling Services" (Integration Broker Administration)

"Setting Service and Schema Namespaces" (Integration Broker Administration)

Setting XML Target Locations

To set the XML target locations, use the Target Locations page (IB_SVCSETUP_SEC). To access the page, on the Service Configuration page, click the **Setup Target Locations** link.

The following example shows the Target Locations page.



The steps for specifying the XML target location are described elsewhere in the product documentation.

See “Understanding Configuring PeopleSoft Integration Broker for Handling Services” (Integration Broker Administration).

Related Links

“Understanding Configuring PeopleSoft Integration Broker for Handling Services” (Integration Broker Administration)

“Setting Target Locations for Services” (Integration Broker Administration)

Chapter 4

Navigating in the Document Builder

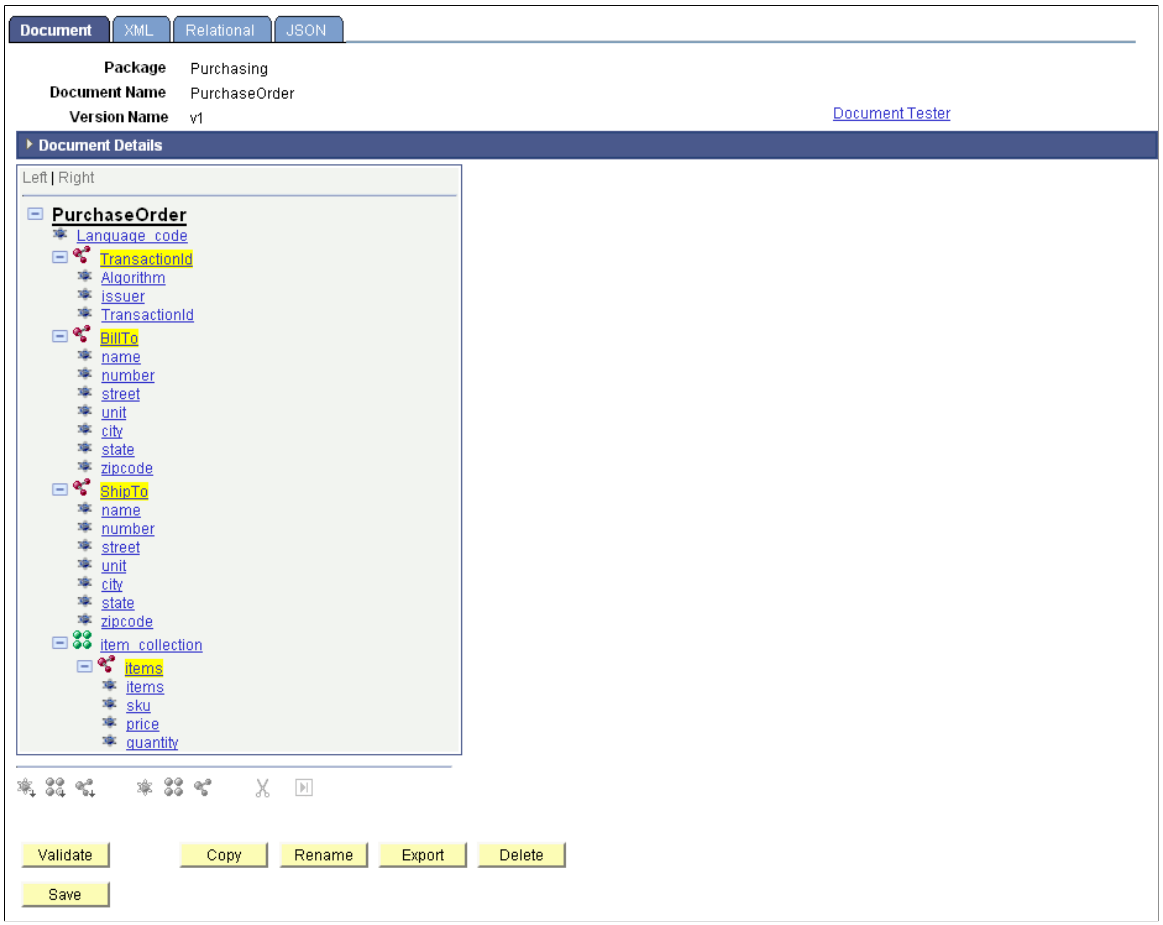
Accessing the Document Builder

PeopleSoft provides a Document Builder tool that enables you to build the structure of a document and define XML and relational formats for it.

The Document Builder is located in the IB_LOGICALSCHEMA component in the PeopleSoft Pure Internet Architecture.

To access the Document Builder, select **PeopleTools** > **PeopleTools** > **Documents** > **Document Builder.** When the Document Builder opens, the default view is of the Document Builder - Documents (IB_LOGICALSCHEMA) page.

This example shows the Document Builder – Documents page, the default view of the Document Builder.



The Document Builder has four pages. The following table briefly describes the pages and how to access each of them.

Page	*Object ID*	*Description*	*Navigation*
Document Builder - Document	IB_LOGICALSCHEMA	Use the Document page to view: • Metadata objects that reference the current document. • General document details, such as comments about the definition, the owner of the object, and more. • A list of documents of which the current document is a child. You also use the Document page to: • Build out the logical structure of the document by adding elements to documents and defining element properties. • Copy, rename, export, and validate a document. Validate the XML document structure.	**PeopleTools** > **Documents** > **Document Builder.** Click the Document tab.
Document Builder - XML	IB_XMLSCHEMA	Use the XML page to view: • XML schema details, such as target and imported namespace information. • Options such as element whitespace, filter blank elements, and more.	**PeopleTools** > **Documents** > **Document Builder.** Click the XML tab.
Document Builder - Relational	IB_RELATSCHEMA	Use the Relational page to map PeopleSoft records to documents, which enables you to populate a rowset with a document.	**PeopleTools** > **Documents** > **Document Builder.** Click the Relational tab.
Document Builder - JSON	IB_JSONSCHEMA	Use the JSON page to view or specify JSON tag names for document elements.	**PeopleTools** > **Documents** > **Document Builder.** Click the *JSON* tab.

Subsequent sections in this topic detail how you navigate each of the Document Builder pages.

Related Links

Navigating the Document Page
Navigating the XML Page
Navigating the Relational Page

Using the Document Tree to View Document Information

The system displays the structure of a document in a hierarchical format in a document tree.

The document tree appears on each page of the Document Builder. In most cases, when you click an element name, additional information appears for you to view or define on the right side of the page. The information that appears depends on the Document Builder page that you are viewing.

This section discusses how to:

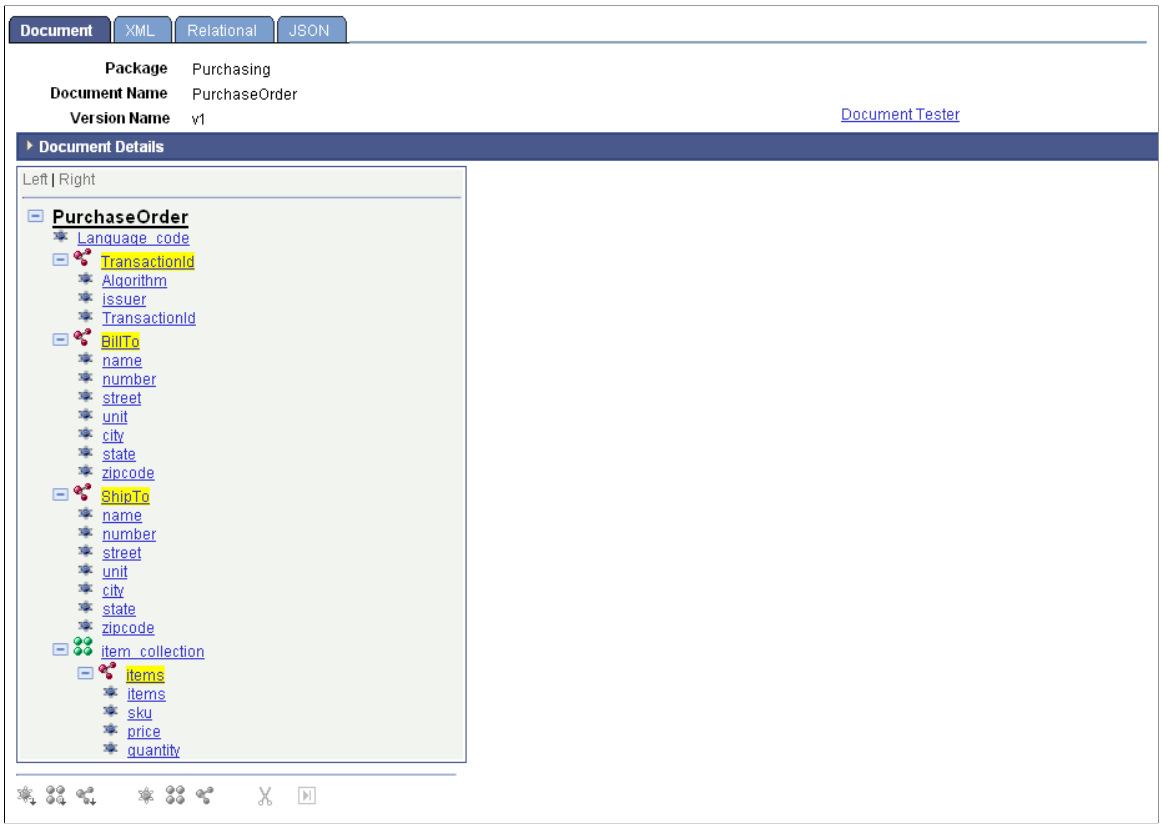
- View the document tree.
- View elements in the document tree.
- Expand and collapse the document tree.
- Select elements and view element properties.

Other topics discuss using the document tree to build and manage documents.

Viewing the Document Tree

When you open a document in the Document Builder, the Document Builder - Document page appears, and the document tree for the definition appears after the Document Details section.

This example illustrates the Document Builder - Document page and shows the *PurchaseOrder* document in the document tree.



Viewing Elements in the Document Tree

A document has one root element and, by default, the same name as the document definition. The root can have one or more child elements; child elements can have none, one, or more peer elements or child elements; and so on.

In the previous example, the root element has the name of the document, *PurchaseOrder.* The root has the following child elements: *Language_Code, TransactionID, BillTo, ShipTo,* and *Item_collection.* All elements except for *Language_Code* have child elements.

When you view elements in the document tree, the icons to the left of the element name identify the element type:

Field or Control	***Description***
	Primitive element.
	Collection element.
	Compound element.

These icons also appear on the buttons of the action toolbar. You use the element action toolbar to add elements to the document tree.

Element types and the action toolbar are discussed in greater detail elsewhere in the product documentation.

Related Links

Understanding Element Types and Data Types
Using the Elements Action Toolbar

Expanding and Collapsing the Document Tree

You can expand and collapse the document tree by clicking the **Expand** (+) and **Collapse** (–) buttons next to the parent elements.

Selecting Elements and Viewing Element Properties

In most cases, when you select an element name in the document tree, properties that are defined or can be defined for the element appear on the right side of the page. This behavior is consistent for all pages of the Document Builder.

This example shows the Document Builder - XML page with the *BillTo* document in the document tree.

The *name* element is selected in the document tree. When you select the element on this page, the system displays XML property information for the element.

This example illustrates the Document Builder - Relational page for the *BillTo* document.

On this page, when you select the *name* element, the system displays properties for any PeopleSoft fields that are mapped to the element to the right of the document tree.

This example illustrates the Document Builder – JSON page for the *BillTo* document.

When you select *name* element, the system displays the JSON tag name that is defined for the element.

Navigating the Document Page

This section discusses how to:

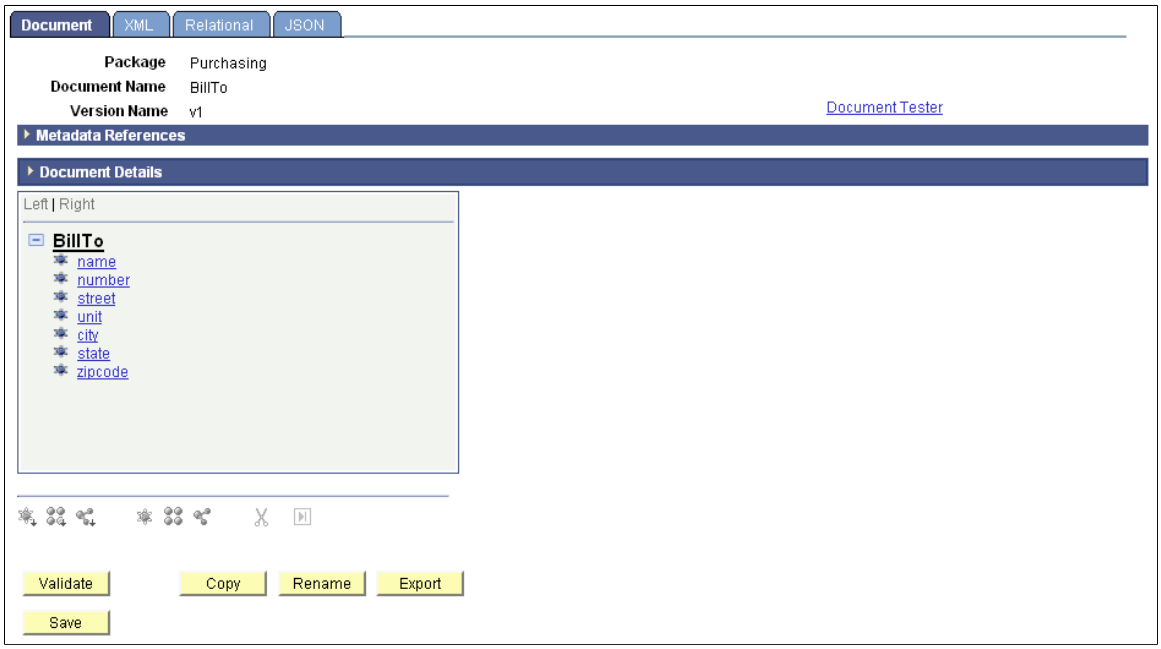
- Access the Document Builder - Document Page
- View metadata references.
- View document details.

- View document dependencies.
- View element properties.
- View the action toolbar.

Accessing the Document Builder - Document Page

The Document page (IB_LOGICALSCHEMA) is the default view in the Document Builder. To access the Document Builder - Document page, select **PeopleTools** > **Documents** > **Document Builder.**

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



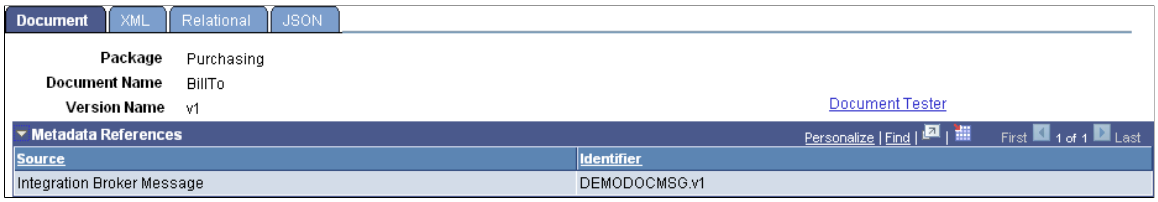
Viewing Metadata References

If a document is referenced by another metadata object in the system, a Metadata References section appears on the Document Builder - Documents page that lists the object type and the object definition name that references the document.

Note: The Metadata References section appears only when another PeopleTools object references the document.

By default, the section appears collapsed on the page. To expand and view the section, click the arrow icon next to the Metadata References section label. To collapse the section, click the arrow again.

The following example shows a partial view of the Document page with the Metadata References section expanded.



The Source column lists the PeopleTools metadata object type that references the document. The Identifier column lists the name of the object definition that references the document.

In the example, an Integration Broker message metadata object references the *BillTo* document. The name of the Integration Broker message referencing the document is *DEMODOCMSG.v1*.

Viewing Document Details

The Document Builder-Documents page features a Document Details section that provides information about the document, such as a document label, owner ID, free - form comments, and whether the document is designated as a private document.

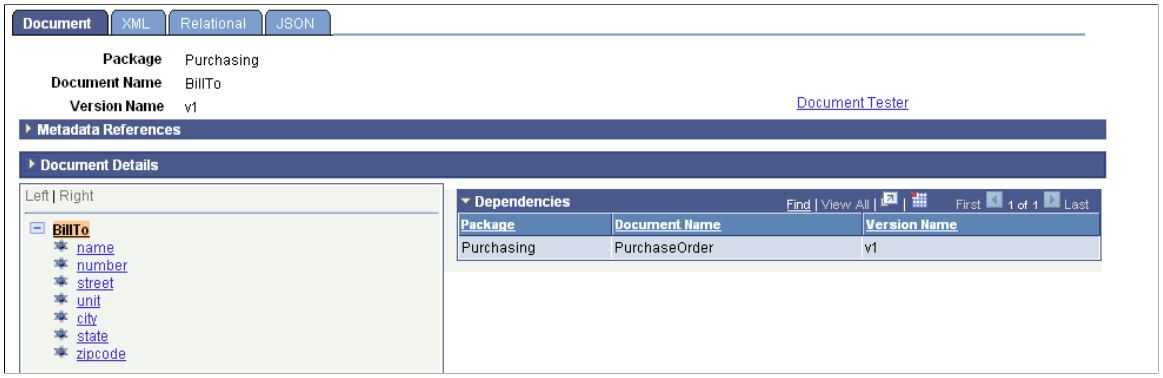
The section appears collapsed by default. To expand and view the section, click the arrow icon next to the Document Details section label. To collapse the section, click the arrow again.

The following example shows the Document Details section for the *BillTo* document.

Viewing Document Dependencies

If a document is a child of another document in the system, then it appears in the **Document Dependencies** grid on the Document Builder - Document page. The **Document Dependencies** grid appears on the right side of the page when you click the root element in the document tree.

The following example shows document dependencies for the *BillTo* document



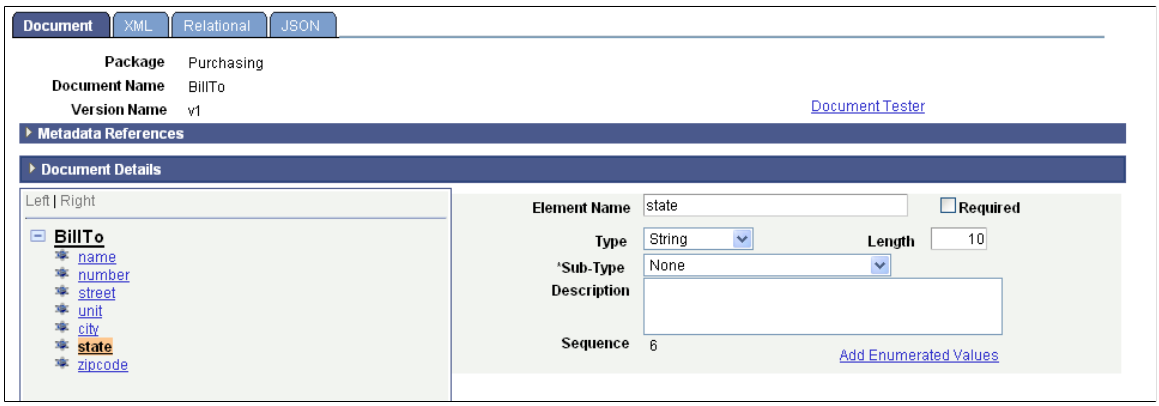
The previous example shows that the *BillTo* document is a child document in the *PurchaseOrder* document.

Unless write-restricted, any changes you make to a document will impact any parent documents.

Viewing Element Properties

When you click any element in the document tree other than the root element, properties for the element appear on the right side of the page.

The following example shows the element properties for the *state* element of the *BillTo* document:



The element properties that appear depend on the element type.

Element properties are discussed in detail elsewhere in the product documentation.

Related Links

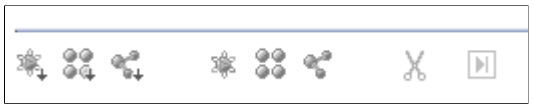
Understanding Element Types and Data Types
Managing Primitive Elements
Managing Compound Elements
Managing Collection Elements

Viewing the Action Toolbar

The action toolbar appears below the document tree on the Document Builder - Document page.

Use the toolbar to add child and peer elements to a document, move elements in the tree, or delete elements in the tree.

The following example shows the action toolbar.



The action toolbar is not enabled until you select an element in the document tree. In addition, the actions that are available to perform on an element vary, depending on the element with which you are working and the structure of the tree.

Using the action toolbar is discussed in more detail elsewhere in the product documentation.

See Using the Elements Action Toolbar.

Navigating the XML Page

This section provides an overview of navigating the XML page and discusses how to:

- Access the Document Builder - XML page.
- View XML schema details.
- View XML schema properties for elements.

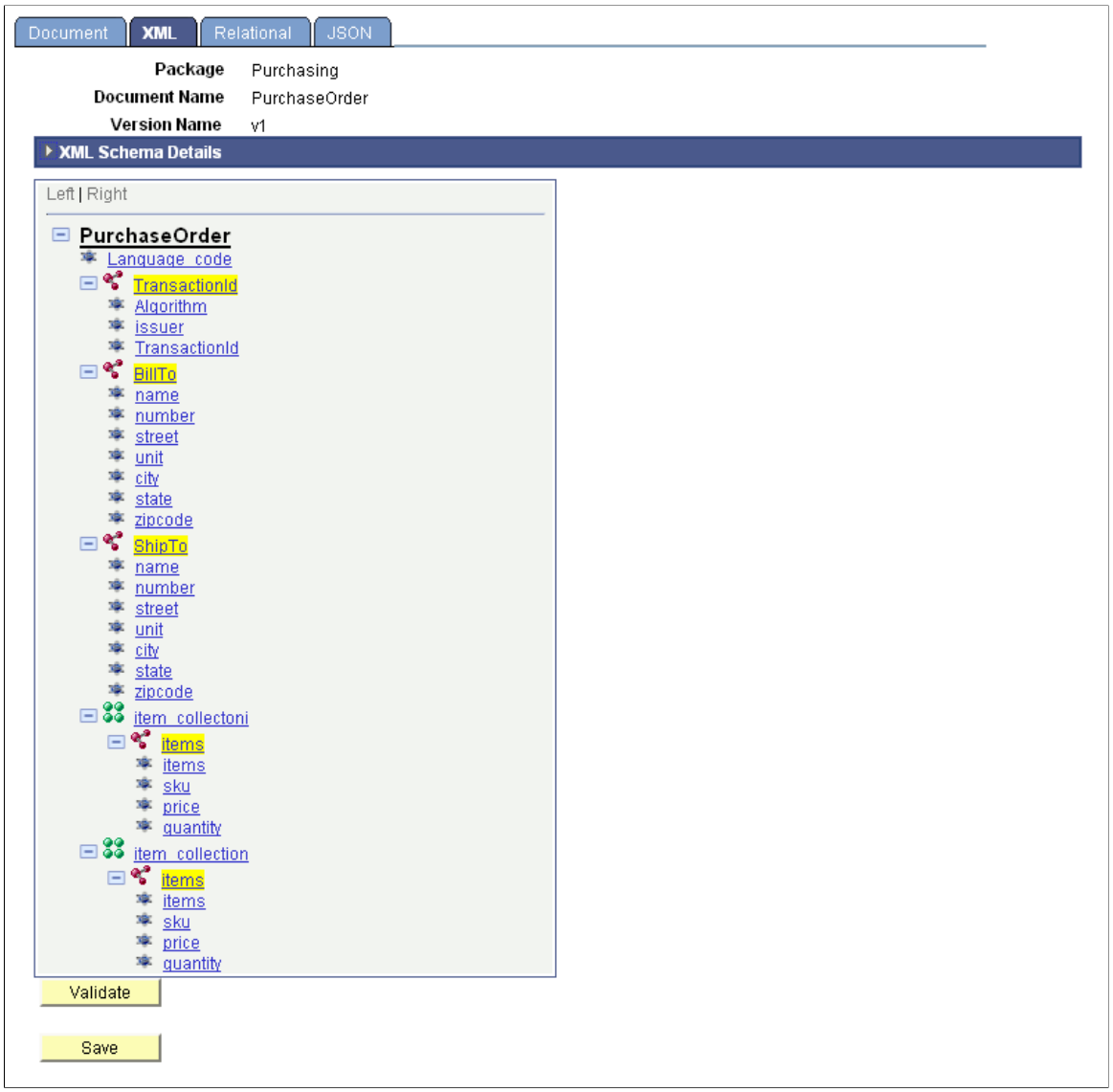
Understanding Navigating the XML Page

The Document Builder - XML page (IB_XMLSCHEMA) features an XML Schema Details section that displays XML schema information defined for the document. It also enables you to view and define schema properties for elements, such as tag names, tag prefixes, and so on.

Accessing the Document Builder - XML Page

To access the Document Builder - XML page, select **PeopleTools** > **Documents** > **Document Builder,**and click the **XML** tab.

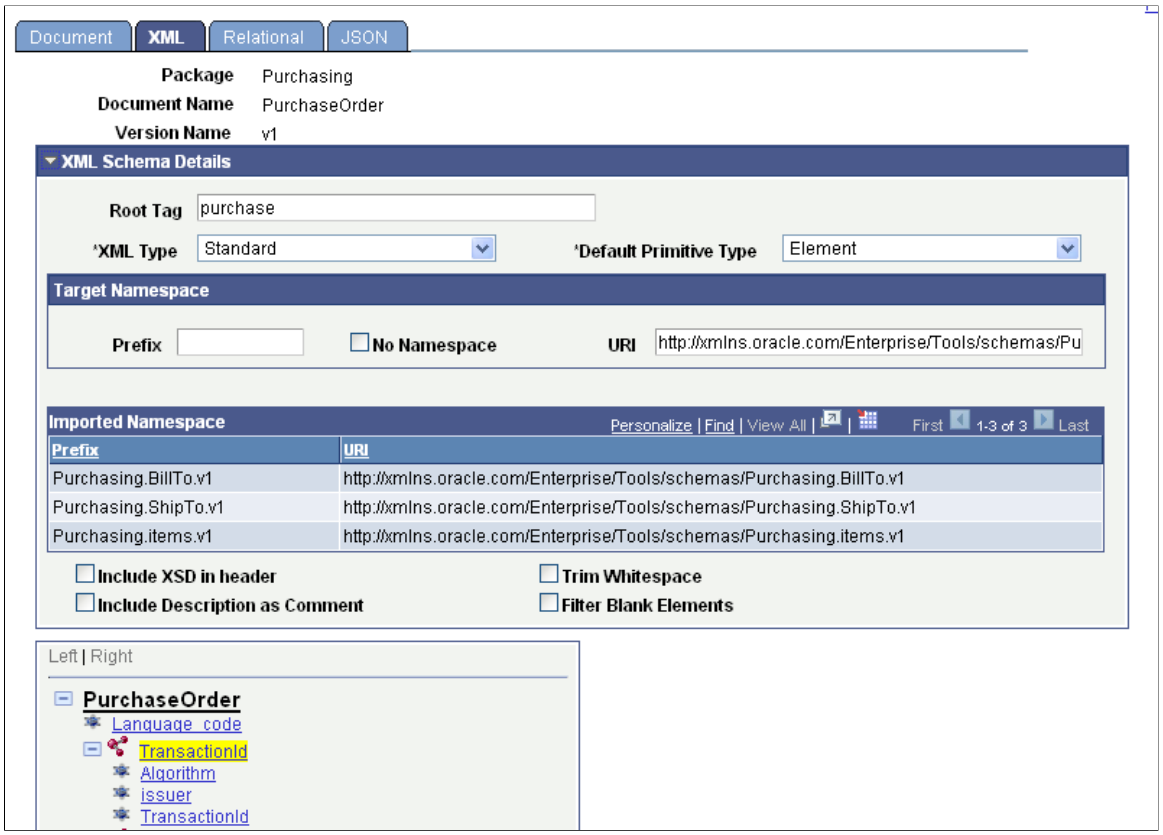
This example illustrates the fields and controls on the Document Builder - XML page. You can find definitions for the fields and controls later on this page.



Viewing XML Schema Details

By default, the XML Schema Details section of the Document Builder - XML page appears collapsed on the page. To expand the section, click the arrow icon next to the XML Schema Details label.

The following example shows the Document Builder - XML page with the XML Schema Details section expanded.

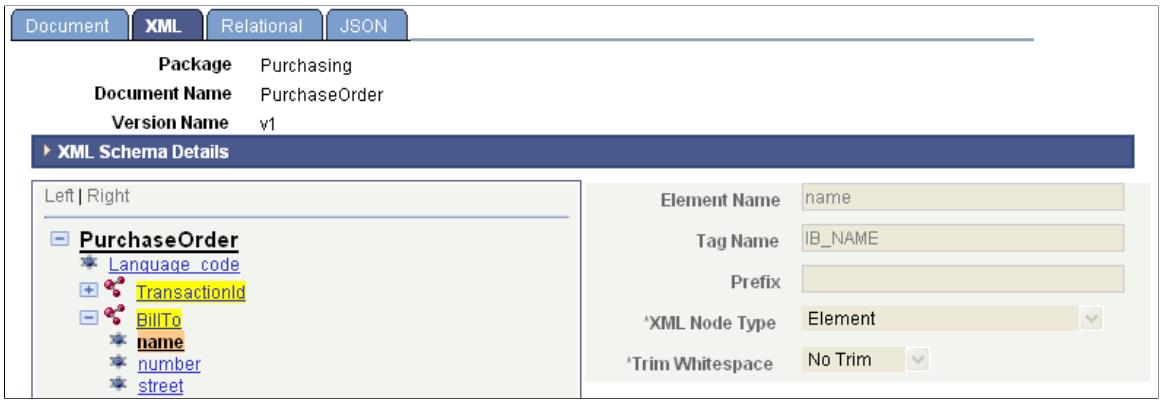


The XML Schema details shows the root tag name, XML type, and element type. Target namespace information provided includes the target namespace prefix and URI,, and if the namespace is defined to appear in the generated XML. The Imported Namespace section shows the names and URIs of any documents that are defined as elements in the document. Other information provided includes whether the schema reference is included in the XML, if whitespace is to be trimmed, if blank elements are filtered out of the XML, and so on.

Viewing XML Schema Properties for Elements

The Document Builder - XML page also displays property information about elements. When you select an element in the document tree, properties for the element appear on the right side of the page.

The following example shows the Document Builder – XML page. The *name* element is selected in the document tree, and the XML schema properties defined for it appear on the right side of the page:



The schema properties for the name element are read-only. They are read-only because the *BillTo* element is actually a document itself. To make any changes to it, you would access the *BillTo* document and make the appropriate changes.

Navigating the Relational Page

This section provides an overview of the Relational page and discusses how to:

- Access the Relational page.
- View records mapped to relational documents.
- View record fields mapped to relational document elements.

Understanding the Relational Page

Use the Document Builder - Relational page (IB_RELATSCHEMA) to view PeopleSoft records and fields that are mapped to relational documents and relational document elements.

Mapping PeopleSoft records and fields to documents is optional.

Usually, mapping PeopleSoft records to documents is done so that you can populate a rowset using a document.

Accessing the Relational Page

To access the Document Builder - Relational page, select **PeopleTools** > **Documents** > **Document Builder** and click the **Relational** tab.

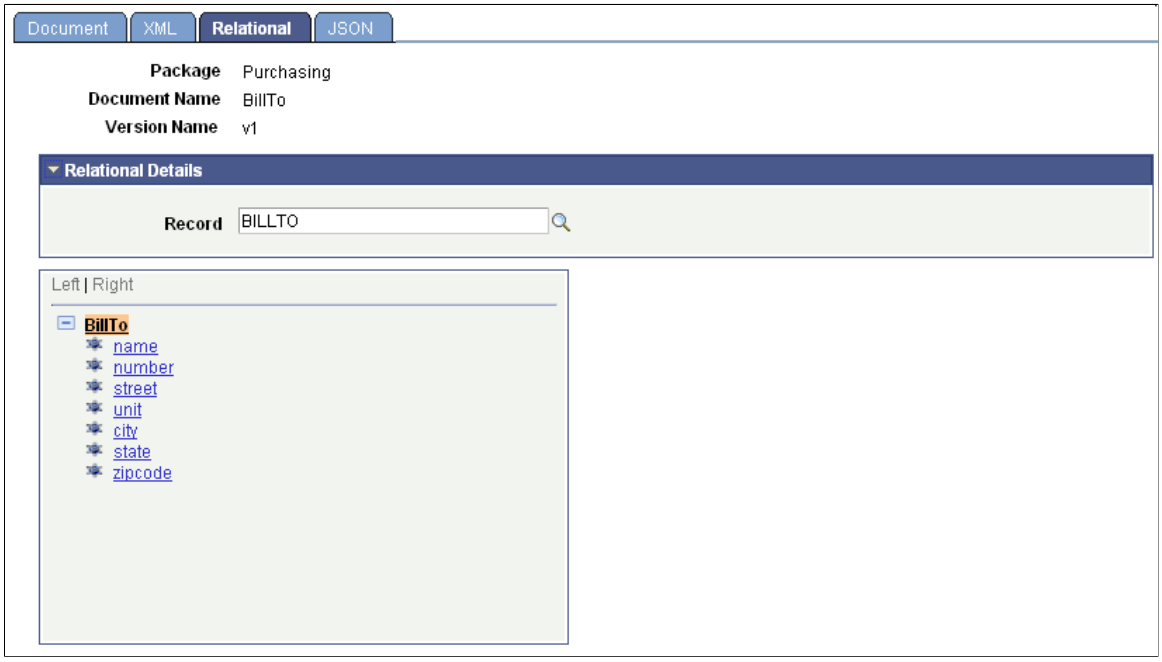
This example illustrates the fields and controls on the Document Builder - Relational page.

Viewing Records Mapped to Relational Documents

The Document Builder - Relational page features a Relational Details section where you can view the PeopleSoft record mapped to the document.

By default, the Relational Details section is collapsed. To expand the section, click the arrow icon next to the Relational Details label.

The following example shows the Document Builder – Relational page with the Relational Details section expanded.



The previous example shows that the PeopleSoft record *ITEMS* is mapped to the *Line_Items* document.

When a PeopleSoft record is mapped to a document, all the elements in the document must be mapped to fields in the record.

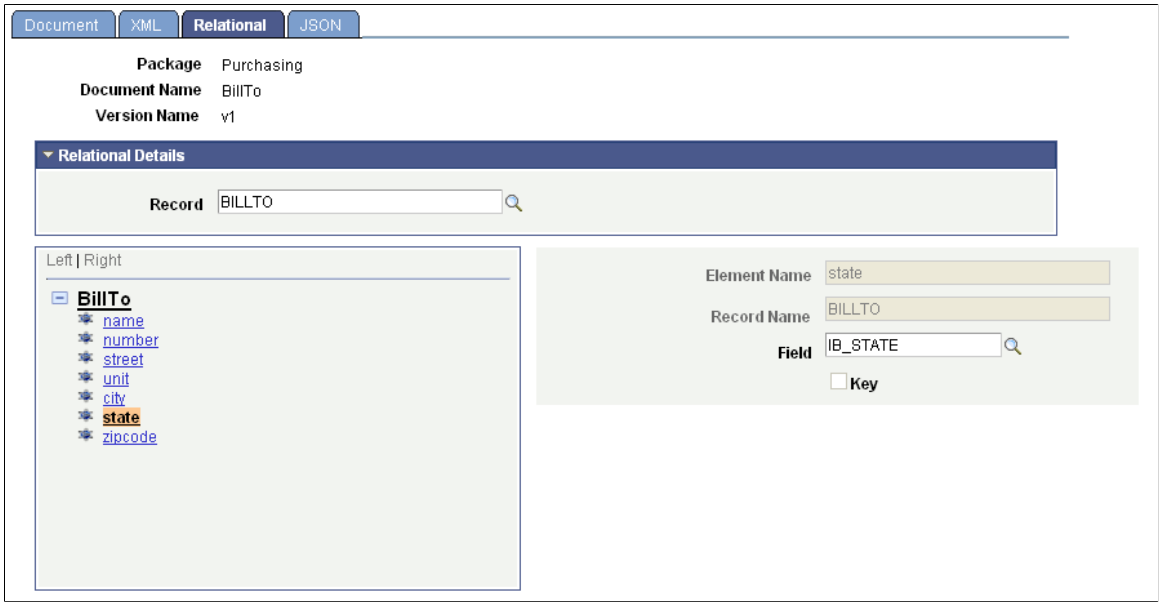
The following section describes how to view record fields that are mapped to document elements.

Viewing Record Fields Mapped to Document Elements

As described in the previous section, if a PeopleSoft record is mapped to a document, then all document elements must be mapped to a field in the record.

To view the record fields that are mapped to document elements, you select an element name in the document tree, and the field mapped to the element appears to the right of the document tree.

The following example shows the mapping of a record field to a document element.



Navigating the JSON Page

This section provides an overview of the JSON page and discusses how to:

- Access the JSON page.
- View the root label for a JSON document.
- View element tag names for a JSON document.

Understanding the JSON Page

Use the Document Builder - JSON page (IB_JSONSCHEMA) to view the root label and JSON objects and arrays that are mapped to documents and document elements.

Accessing the JSON Page

To access the JSON page select **PeopleTools** > **Documents** > **Document Builder** and click the **JSON** tab.

Use the Document Builder – JSON page to view document elements mapped to JSON documents.

Viewing Root Labels for JSON Documents

The Document Builder – JSON page features a JSON Details section that displays the JSON root label. By default the label is populated with the root label of the logical document.

You can edit the root label as necessary. However, if left blank, the system will add the root label from the logical document at save time.

By default, the JSON Details section is collapsed. Click the arrow icon that appears next to the label to expand or collapse the section.

The previous example of the Document Builder – JSON page shows the JSON Details section expanded. The **Root Label** field is populated with the name of the root element of the document, which is the default value.

Viewing Element Tag Names for JSON Documents

When you click the name of an element in the document tree on the Document Builder – JSON page, the element name appears in read-only format on the right side of the page. In addition, the JSON tag name appears in the **JSON Tag Name** field.

The JSON tag name is the name that will appear for the element in the generated document or when the document data is parsed. By default, JSON tag name for an element is the element name from the logical document. However, you can edit the value as desired.

The previous example of the Document Builder – JSON page shows the name element selected in the document tree. On the right side of the page, the element name appears in a read-only **Element** field. The JSON tag name that has been defined for the element, *IB_NAME,* appears in the **JSON Tag Name** field.

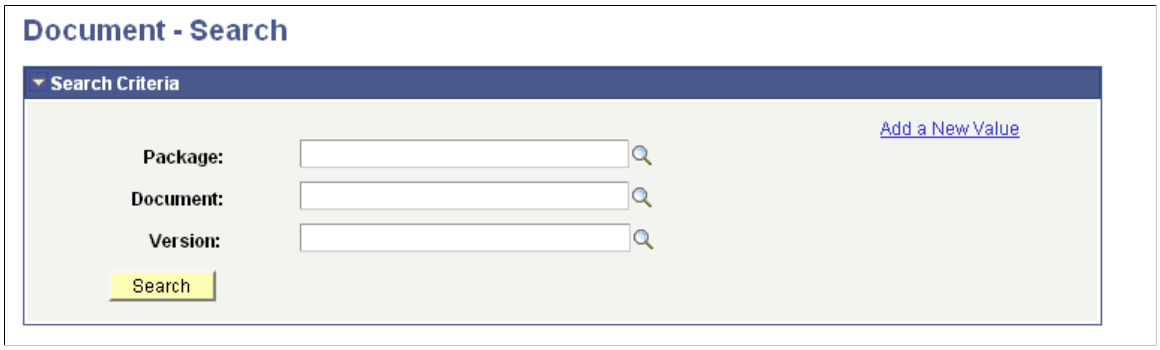
Chapter 5

Searching For and Adding Documents

Searching for Document Definitions

Use the Document – Search page (IB_MSGSEARCH) to locate document definitions in the database. To access the Document – Search page, select **PeopleTools** > **Documents** > **Document Builder.**

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



You can locate a document in the system by searching by one or more of the following attributes: package name, document name, and version number. To specify an attribute, enter the value in the corresponding field or click the **Lookup** button to search from all values in the database.

To search for documents:

1. Access the Document Builder search page (**PeopleTools** > **Documents** > **Document Builder**).

   The search page appears.

2. Enter a value in one or more of the following fields:

   a. In the **Package** field, enter the package name or click the **Lookup** button to search for a value.

   b. In the **Document** field, enter the document name or click the **Lookup** button to search for a value.

   c. In the **Version** field, enter the document version or click the **Lookup** button to search for a value.

   Note: To display all document definitions in the database, leave all fields blank and click the **Search** button.

3. Click the **Search** button.

   The search results appear in the **Search Results** grid at the bottom of the page.

4. In the **Search Results** grid, locate and click the document with which to work.

The document definition appears in the Document Builder.

Adding Document Definitions

This section provides information about naming documents and discusses how to add document definitions to the system.

Understanding Naming Document Definitions and XML Root Tag Names

By default, the document name you specify when you add a document to the system also becomes the name of the XML root tag that appears in XML document schema. However you can specify a different root tag name on the XML tab in the XML Schema Details section in the Root Tag field. The character field limit for document names is 100 characters.

The character field limit for root tag names is 30 characters.

If the document name exceeds 30 characters, the system automatically populates the Root Tag field with only the first 30 characters of the document name. However, the full document name, up to the document name character limit of 100, appears in the generated XML document schema. The exception is if you modify the Root Tag field; if you choose to do so, that value is used as the root tag element in generated schema.

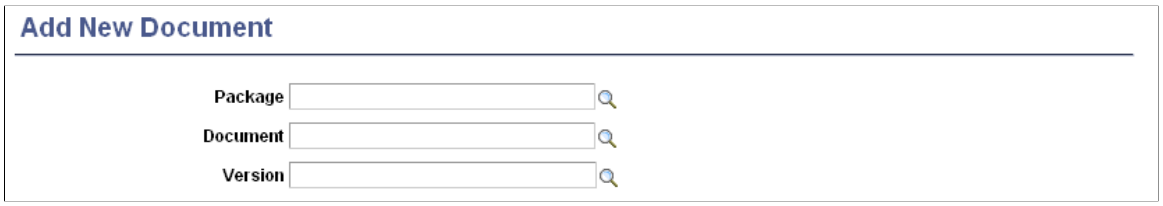
Related Links

Defining XML Schema Details

Adding Documents

Use the Document Builder – Add New Document page (IB_MSGSEARCH_ADD) to add a new document to the database. To access the page, select **PeopleTools** > **Documents** > **Document Builder.** The Document – Search page appears. Click the **Add a New Value** link.

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



After you complete naming the document, you must click the **Add** button and add at least one child element to the document before you can save it in the database.

To add a document definition:

1. Access the Add New Document page (**PeopleTools** > **Documents** > **Document Builder and click the Add a New Value link**).

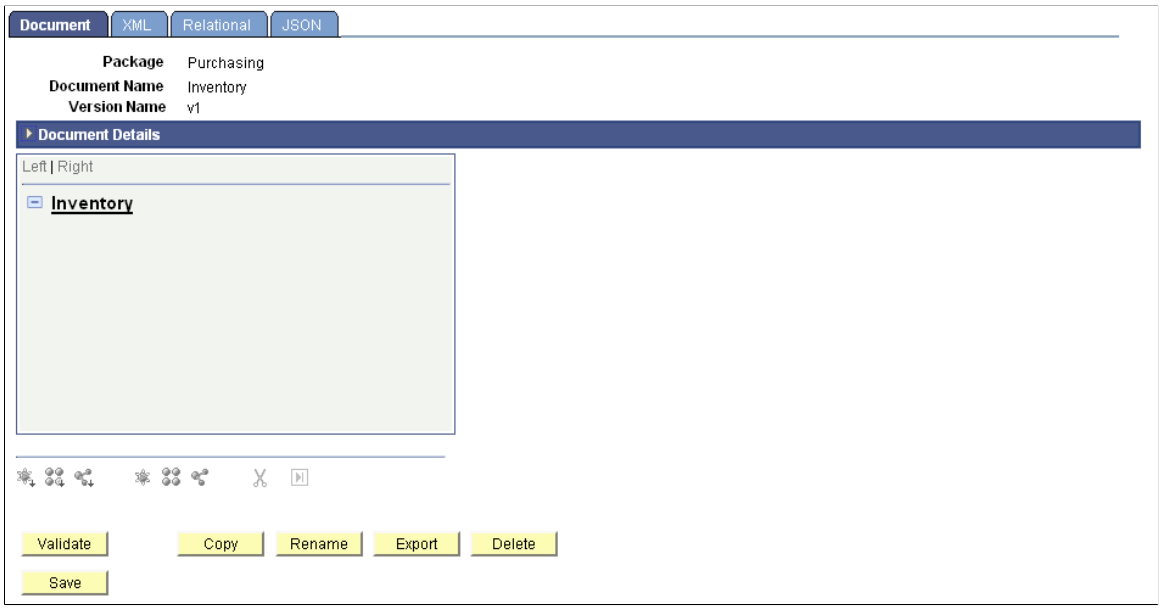
The Add New Document page appears.

2. In the **Package** field, enter a package name.
3. In the **Document** field, enter a document name.

   Note that the value you enter in this field becomes the root element name for the document.
4. In the **Version** field, enter a document version.
5. Click the **Add** button.

The document definition appears in the Document Builder - Document page.

This example shows the Document Builder – Document page for a new document definition called *Inventory.*



When you create a new document, the root element appears in the document tree. However, the definition is not yet saved in the database. You must add at least one child element to the document before you can save it in the database.

Chapter 6

Adding Elements to Documents

Understanding Element Types and Data Types

This section describes the element types and element data types that the PeopleSoft system supports as part of its document framework.

Note: The term and concept of elements described in this topic refers to node elements.

Element Types

When you manage documents in the PeopleSoft system, you can add and manage root elements, child elements, and peer elements.

The first element that appears in a document is the root element. A document can have only one root element. When you create a new document, the system automatically creates a root element and uses the document name as the root element name.

Data Types

This section discusses the data types that PeopleSoft supports.

Term	*Definition*
Primitive	PeopleSoft supports the primitive data types in the following list. You may use primitive data types in child and peer elements. • Binary • Boolean • Character • Date • DateTime • Decimal • Integer • String • Text • Time **Note:** When an Integer or Decimal is selected as a primitive and the unbounded checkbox is selected then the maximum that will be allowed is a length of 31. This maximum includes the negative ,if applicable, and the fractional part for decimals.
Compound	A compound data type can be a PeopleSoft record, complex primitive data type, or another document.
Complex primitive	A complex primitive data type is a primitive data type that can contain attributes. With this data type, the system-generated XML creates the element tag with the associated attributes as a sibling instead of a child.
Collection	Collection elements represent multiple occurrences of a single type, similar to an array or a set.

Related Links

Defining Primitive Element Properties
Managing Compound Elements
Managing Collection Elements

Using the Elements Action Toolbar

These topics provide an overview of the elements action toolbar discuss how to:

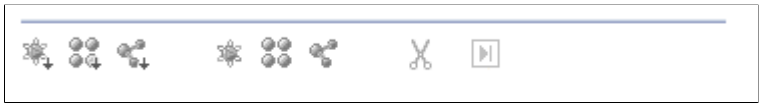
- Access and enable the action toolbar.
- Use the action toolbar buttons.

Accessing and Enabling the Action Toolbar

Use the action toolbar to select element types and data types to add to a document.

The elements action toolbar appears on the Document Builder - Documents page under the document tree.

This example shows the action toolbar.

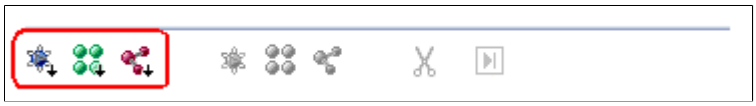


Hover over any button to display the element type that you can add to the system

Child Element Action Buttons

There are three buttons on the action toolbar that you can use to add child elements to a document.

The first set of three buttons highlighted in the following example enable you to add child element types to a document.



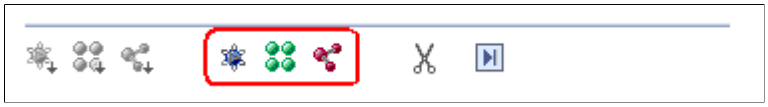
The following table lists the child element buttons in the order they appear on the toolbar and the action you can perform with each:

Term	***Definition***
	Add primitive child.
	Add collection child.
	Add compound child.

Peer Element Action Buttons

Use the element action toolbar to add peer element types to a document:

The second set of three buttons highlighted in the following example enable you to add peer element types to a document.



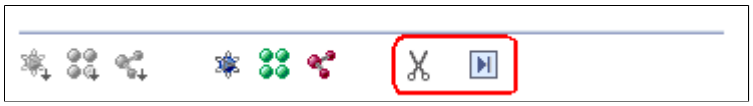
The following table lists the peer element action buttons in the order they appear on the toolbar and the action you can perform with each:

Term	***Definition***
	Add primitive peer.
	Add collection peer.
	Add compound peer.

Delete and Move Element Action Buttons

Use the element action toolbar to delete elements from the document tree and move elements in a document tree.

The following example highlights the action toolbar buttons you can use to delete and move elements in the document tree.



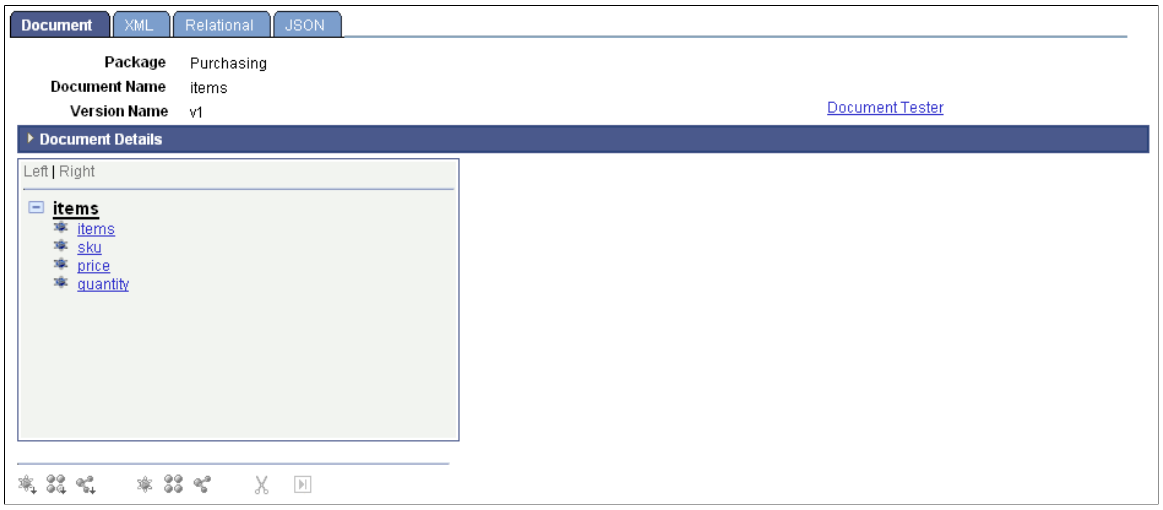
The following table describes the action buttons you can use to delete and move elements in the document tree:

Term	***Definition***
	Delete element.
	Move element to the bottom of the tree.

Accessing and Enabling the Action Toolbar

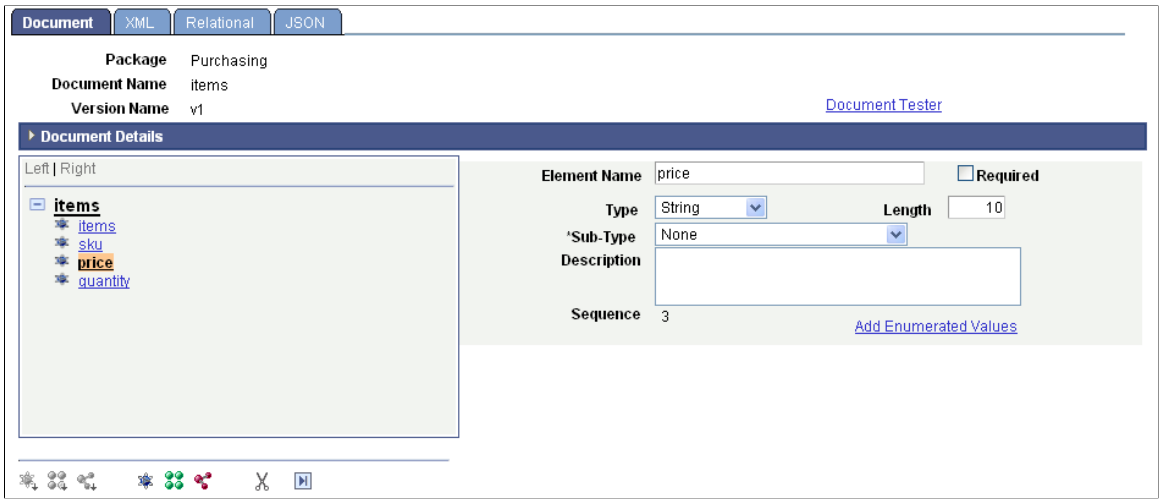
To access the action toolbar, select *PeopleTools, Documents, Document Builder* and then select a document or create a new document.

The following example shows the action toolbar appearing under the document tree section on the Document Builder – Document page



The action toolbar is not enabled until you click an element in the document tree. In the previous example, the action toolbar is not enabled since no element is selected.

This example shows the action toolbar enabled. The toolbar is enabled because an element in the document tree, *price,* is selected.



When you click an element, the element types and data types that you can add are enabled on the toolbar. If an element or data type is not enabled on the toolbar, either you have not selected an element to which to add an element or you are not allowed to add the element type or data type to the document tree at that time.

When you select the root element on the document tree, the Dependencies section appears. Document dependencies are discussed elsewhere in the product documentation.

See Viewing Document Dependencies.

When you select any other element type on the document tree, the properties defined for the element appear on the right side of the page, as shown in the previous example. Defining element properties is discussed elsewhere in this topic.

Related Links

Defining Primitive Element Properties
Managing Compound Elements
Managing Collection Elements

Managing Primitive Elements

These topics provide an overview managing primitive elements and discuss how to:

- Add primitive elements.
- Define primitive element properties.
- Add enumerated values.

Understanding Adding Elements to Documents

You build out a document structure by adding elements to a document tree.

Use the action toolbar on the Document Builder - Documents page to add elements to documents. As described elsewhere in this topic, the Action Toolbar features buttons to add primitive, compound, and collection elements, both child and peer elements, to documents.

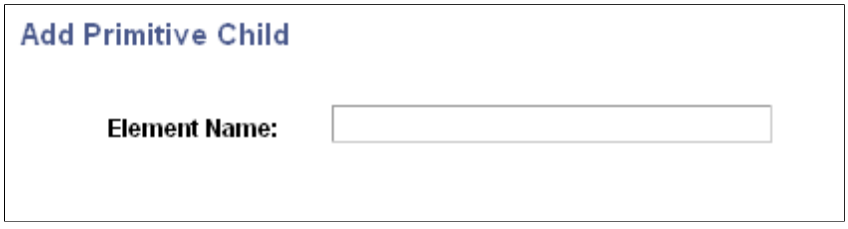
Note: After you add an element to a document, you must define the element properties before you can save the document.

Adding Primitive Elements

To add a primitive element to a document, on the document tree you select the element to which to add a primitive child or primitive peer element. On the Action Toolbar you click the Add Primitive Child or Add Primitive Peer button.

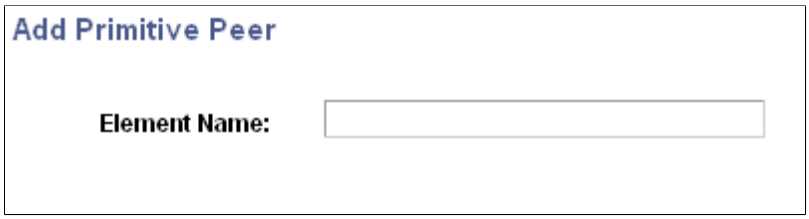
If you are adding a primitive child element, the Add Primitive Child page (IB_LOGICALCOLL_SEC) appears. Use the page to name the new primitive child element. :

The following example shows the Add Primitive Child page.



If you are adding a primitive peer element, the Add Primitive Peer page (IB_LOGICALCOLL_SEC) appears. Use the page to name the new primitive peer element.

The following example shows the Add Primitive Peer page.



Note: The Add Primitive Child and Add Primitive Peer pages use the same IB_LOGICALCOLL_SEC object.

To add primitive elements (child or peer) to documents:

1. Access the Document Builder - Documents page (**PeopleTools** > **Documents** > **Document Builder**).
2. On the document tree, click the element to which to add a primitive child or primitive peer element.
3. On the action toolbar, do one of the following:
   - Click the **Add Primitive Child** button to add a primitive child element.
   - Click the **Add Primitive Peer** button to add a primitive peer element.

   If adding a primitive child element, the Add Primitive Child page appears; if adding a primitive peer element, the Add Primitive Peer page appears.
4. In the **Element Name** field, enter a name for the new primitive element.
5. Click the **OK** button.

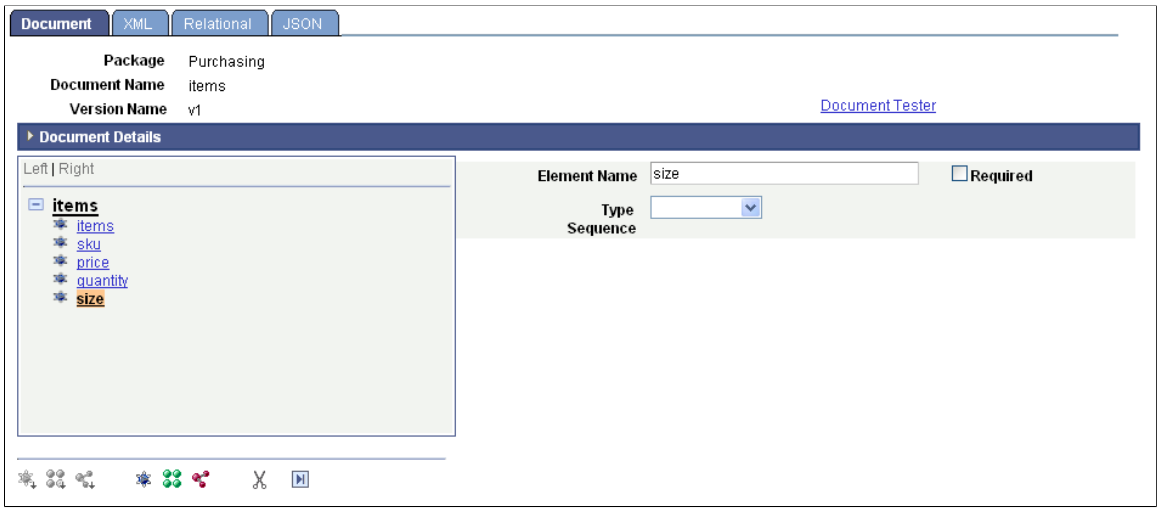
The element appears on the document tree and you may define its properties.

See Defining Primitive Element Properties.

Defining Primitive Element Properties

When you add a primitive element data type to a document, the properties that you can define for the element appear on the right side of the Document Builder - Document page.

In this example, a primitive element data type is selected. The element properties available to define for the new primitive appear to the right of the document tree.



The element data type determines the available properties to define. The primitive data types supported are described elsewhere in this topic.

See Data Types.

The following table describes the properties that you can define for primitive data type elements. Except where noted, the properties in the table are applicable to all primitive data types supported by the system:

Field or Control	***Description***
Add Enumerated Values and Show Enumerated Values	Click the link to define accepted values and their descriptions for a data type. You can define enumerated values for the following primitive data types: Character, Date, DateTime, Integer, String, and Time. See Adding Enumerated Values.
Date/Time Format	From the drop-down list, select one of the following ISO standard date/time format options: • THH:MM:SS.ssssss+/-hhmm • THH:MM:SS.ssssss+/-hh:mm
Decimal Length	Enter the number of digits to include to the right of the decimal point. This property is applicable to the Decimal primitive data type.
Description	Enter a long description of the element.

Field or Control	*Description*
Element Name	Enter the name of the element as it will appear in the XML document.
Length	Enter a length for the element. This property is applicable to the following primitive data types: Binary, Character, Decimal, Integer, String, and Text. **Note:** For a decimal primitive element, the decimal length is included as part of the overall length.
Required	Select the check box to indicate that the element must be included in runtime XML and defined as a required element in the XML schema.
Sequence	The system assigns this read-only field. The sequence number is used for indexing with the PeopleCode API for documents.

Field or Control	*Description*

Sub-Type	Select a value from the drop-down list box to further define the data type.
	
	This property is applicable to the following primitive data types: Integer, String and Text.
	
	The valid values for these data types are:
	
	• Integer.
	  • None. (Default.)
	  • nonNegativeInteger.
	
	• String.
	  • None. (Default.)
	  • QName.
	  • anyURI.
	  • gDay.
	  • gMonth.
	  • gYear.
	  • gYearMonth.
	  • normalizedString.
	  • token.
	
	• Text.
	  • None. (Default.)
	  • QName.
	  • anyURI.
	  • gDay.
	  • gMonth.
	  • gYear.
	  • gYearMonth.
	  • normalizedString.
	  • token.

Field or Control	*Description*
Type	From the drop-down list box, select a primitive data type. The valid values are discussed elsewhere in this topic. See Data Types.
Unbound Maximum	Select the check box to indicate that the length of the element is unlimited. This property is applicable only to the binary primitive data type.

Adding Enumerated Values

You can define allowed values, or *enumerated values,* for some primitive data types.

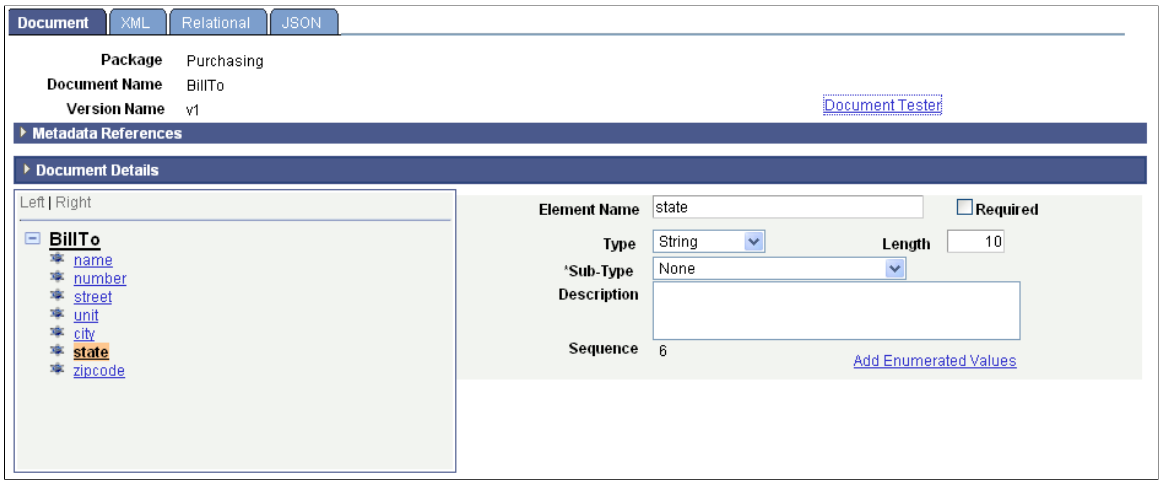
An example of enumerated values is a list of acceptable state abbreviations that can be used in a primitive element called State that is a string.

In the Documents Builder, you can define enumerated values for the following primitive data types:

- Character.
- Date.
- DateTime.
- Integer.
- String.
- Time.

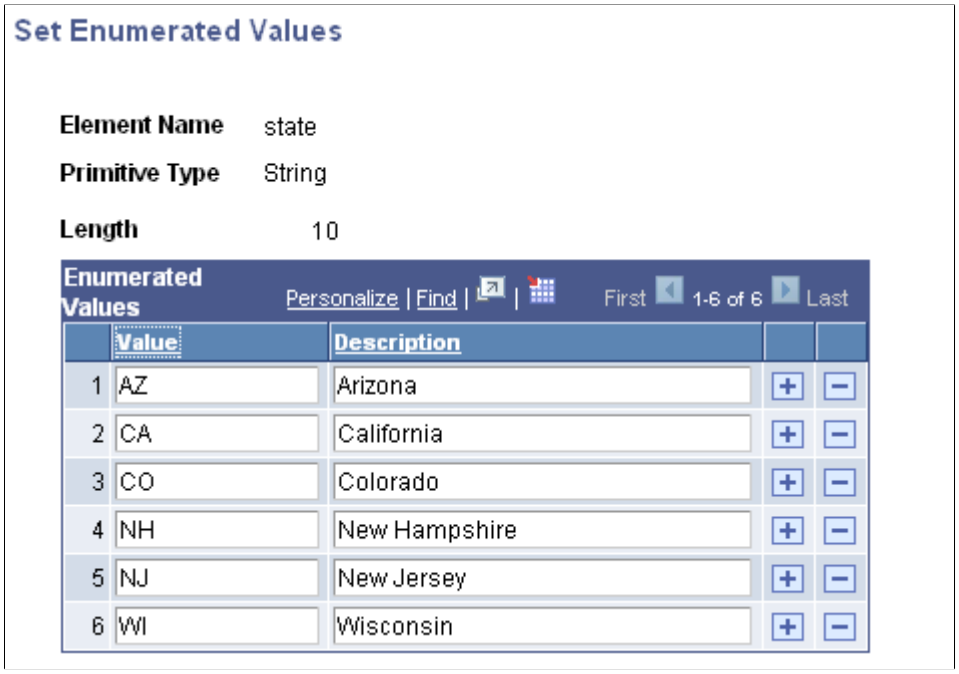
Use the Set Enumerated Values page (IB_ENUM_SEC) to set enumerated values. You access the page by using the **Add Enumerated Values** link that appears on the Document Builder - Documents page when you define properties for one of the primitive data types in the previous list.

The following example shows the **Add Enumerated Values** link that appears when you define a primitive element called *State* that is a string:



Click the **Add Enumerated Value** link to access the Set Enumerated Values page.

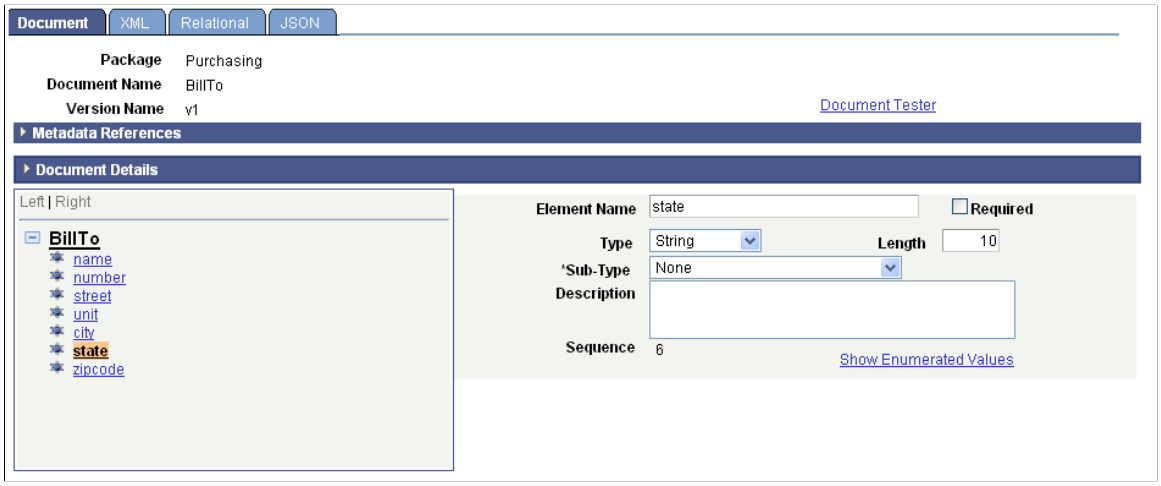
This example illustrates the fields and controls on the Set Enumerated Values page. You can find definitions for the fields and controls later on this page.



To set enumerated values, enter an acceptable value in the **Value** field and then a description for the value in the **Description** field. In the previous example, the Value column shows the acceptable abbreviations and values that the State element can contain for several of the United States. The **Description** field shows information about each of the values.

When enumerated values exist for a primitive data type, when you are working in the properties section a **Show Enumerated Values** link appears instead of a **Set Enumerated Values** link.

The following example shows the properties section for the State element after enumerated values are defined. Note that a **Show Enumerated Values** link appears. Click the link to view or modify defined enumerated values.



To set enumerated values:

1. Select **PeopleTools** > **Documents** > **Documents Builder.**

   The Documents Builder - Documents page appears.

2. In the document tree, select the element for which to add enumerated values.

   The properties for the element appear on the right side of the page.

3. Perform one of the following actions:

   a. Click the **Add Enumerated Values** link to add values.

      This link appears if no enumerated values are defined for the element.

   b. Click the **Show Enumerated Values** link to modify existing values.

      This link appears if enumerated values are already defined for the element.

4. Enter enumerated values and descriptions:

   a. In the **Value** field, enter an acceptable value for the element.

   b. In the **Description** field, enter a description for the value.

5. Click the Add a Row button (+) to add additional rows of values and descriptions.

6. Click the **OK** button.

   The Documents Builder - Document page appears.

Managing Compound Elements

These topics provide an overview of compound data types and discuss how to:

- Add compound elements.
- Define complex primitive data types.
- Define documents as compound element data types.
- Define PeopleSoft records as a compound element data types.

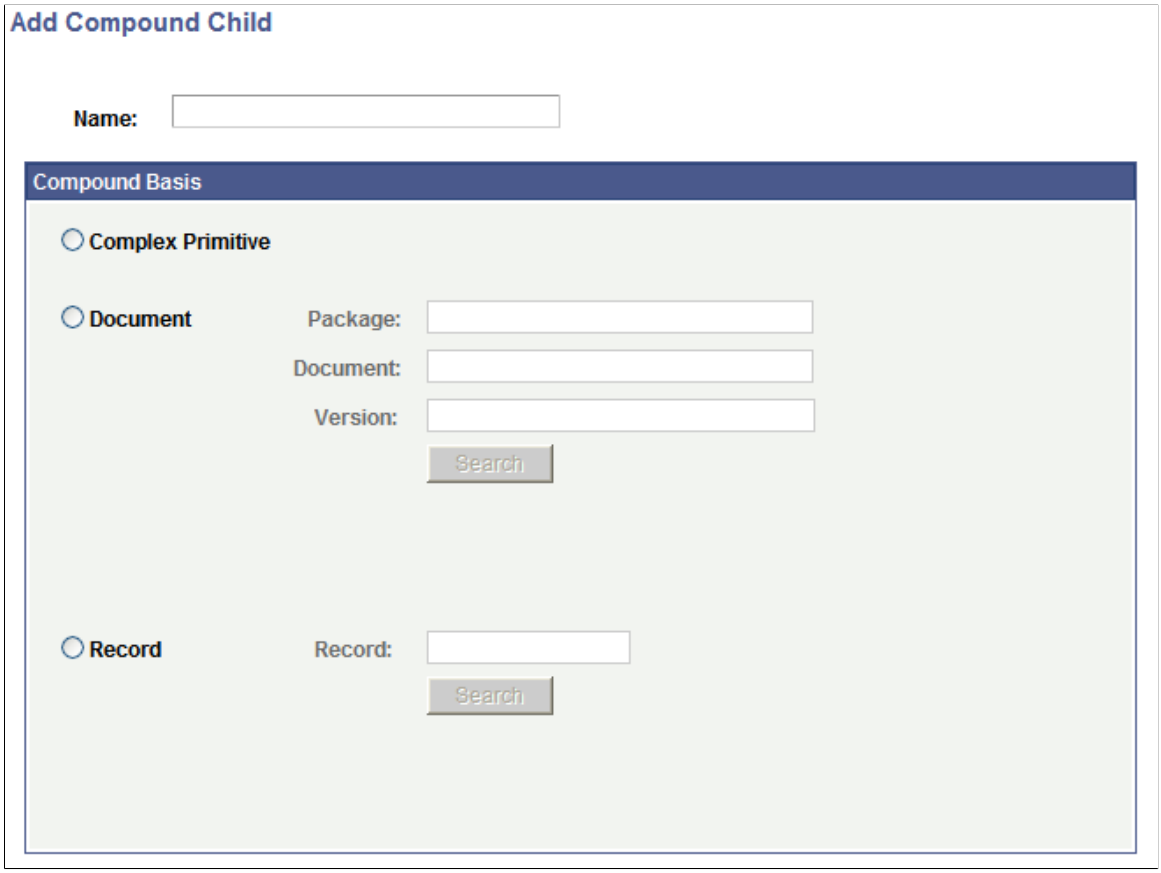
Understanding Defining Compound Data Types

A compound data type can be a complex primitive data type, another document, or a PeopleSoft record.

Adding Compound Elements

When you add a compound data type to the document tree, the Add Compound Child or Add Compound Peer page appears, depending on whether you are adding a child or peer element. The Add Compound Child page and the Add Compound Peer page are identical and have the same object name, IB_LOGICALCOMP_SEC.

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



To add compound elements (child or peer) to documents:

1. Access the Document Builder - Documents page (**PeopleTools** > **Documents** > **Document Builder**).
2. On the document tree, click the element to which to add a compound child or compound peer element.
3. On the action toolbar, do one of the following:
   - Click the**Add Compound Child** button to add a compound child element.
   - Click the **Add Compound Peer** button to add a compound peer element.

     The Add Compound Child page or Add Compound Peer page appears.
4. In the **Name** field, enter a name for the compound element.
5. Select the type of compound element to create. The options are:
   - Complex Primitive

     When you select this option, the Document Builder - Documents page appears for you to define the element properties.

     See Defining Complex Primitive Data Types.
   - Document

     When you select this option, you search for and specify the document to use as the compound element.

     See Defining Documents as Compound Element Data Types.
   - Record

     When you select this option, you search for and specify the PeopleSoft record to use as the compound element.

     See Defining PeopleSoft Records as Compound Element Data Types.

Defining Complex Primitive Data Types

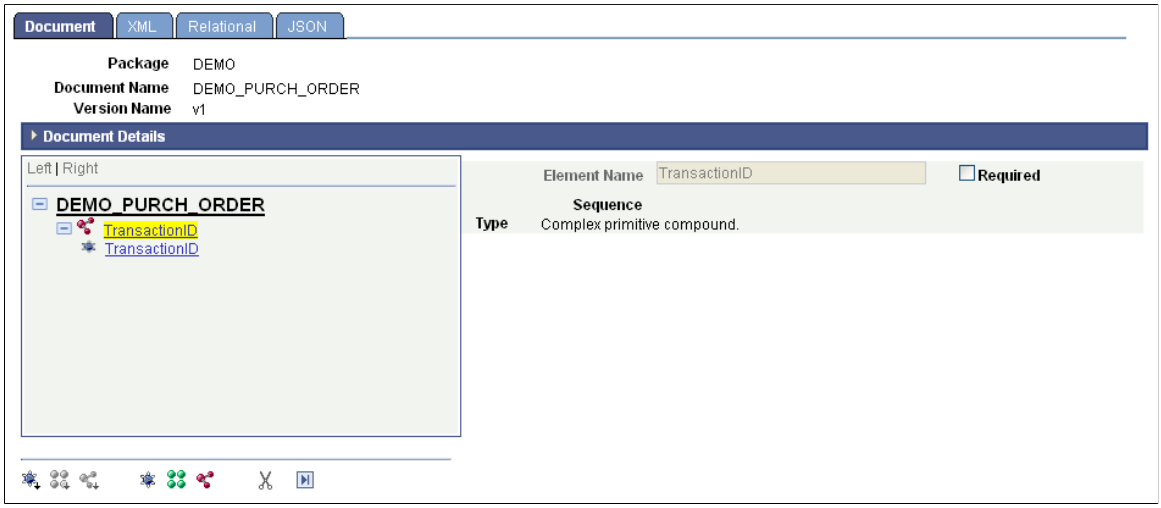
This section discusses how to:

- Define properties for complex primitive elements.
- Define attributes element properties.
- Add additional attribute elements to complex primitive data types.

Defining Properties for Complex Primitive Elements

After you create and name a complex primitive data type, two elements appear in the document tree on the Document Builder - Documents page. One element is for the complex primitive and the other is for the first of one or more element attributes.

The following example shows a newly created complex primitive element (and attribute) element appearing in the document tree. *TransactionID* is a complex primitive element off of the root element.



The complex primitive element is denoted by a complex primitive icon to the left of the element name. In addition, it is highlighted. The attribute is denoted by the primitive icon to the left of the name.

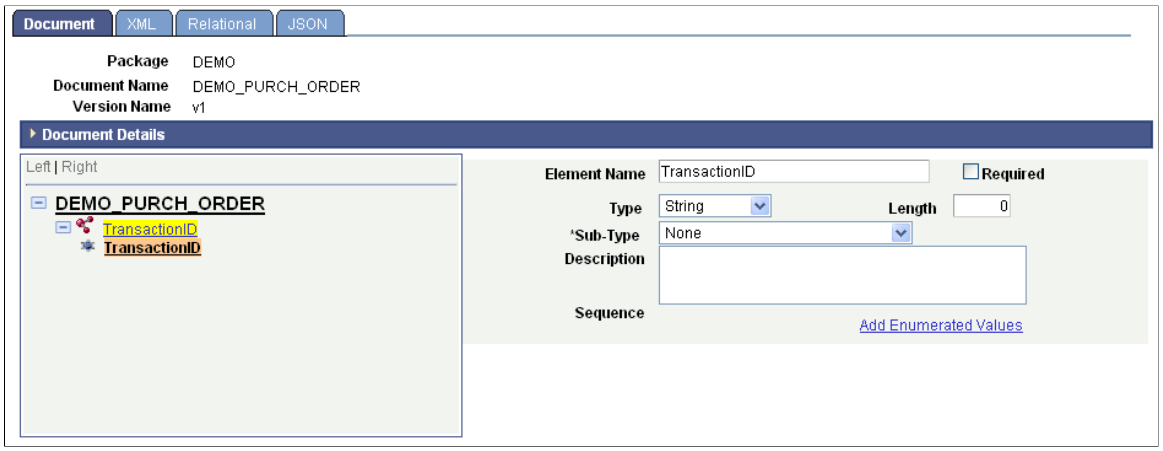
In the example, the complex primitive element is highlighted, and the properties appear on the right side of the page. They are:

Field or Control	***Description***
Element Name	The name of the element as defined on the Add Compound Child or Add Compound Peer page. This value is read-only.
Required	Select the check box to make the element a required element.
Sequence	The system assigns this read-only field. The sequence number is used for indexing with the PeopleCode API for documents.
Type	This field displays the element type.

Defining Attribute Element Properties

Properties you define for an attribute element appear when you select an element in the document tree.

The following example shows the attribute element *TransactionID* selected in the tree.



When you select an attribute element, the properties you can define for it appear on the right side of the page.

To specify attribute element properties:

1. In the document tree, select an attribute element.

   The attribute element properties appear on the right side of the page.

2. In the **Element Name** field, enter the name of the element as you want it to appear in the generated XML.

   If you change this name, the complex primitive element and attribute element name changes to the value you enter.

3. Select the **Required** box to make the attribute a required element.

4. From the **Type** drop-down list box, select an element type.

   Primitive element types are described elsewhere in this topic.

   See Defining Primitive Element Properties.

5. Click the **Save** button.

Adding Additional Attribute Elements to Complex Primitive Data Types

You can add the following element types as attribute elements to complex primitive data types:

- Primitive child.
- Primitive peer.
- Compound peer.
- Collection peer.

You add these elements to complex primitives as you would any other element, first using the action toolbar on the Document Builder - Documents page to add the element to the document tree, and then defining the element properties.

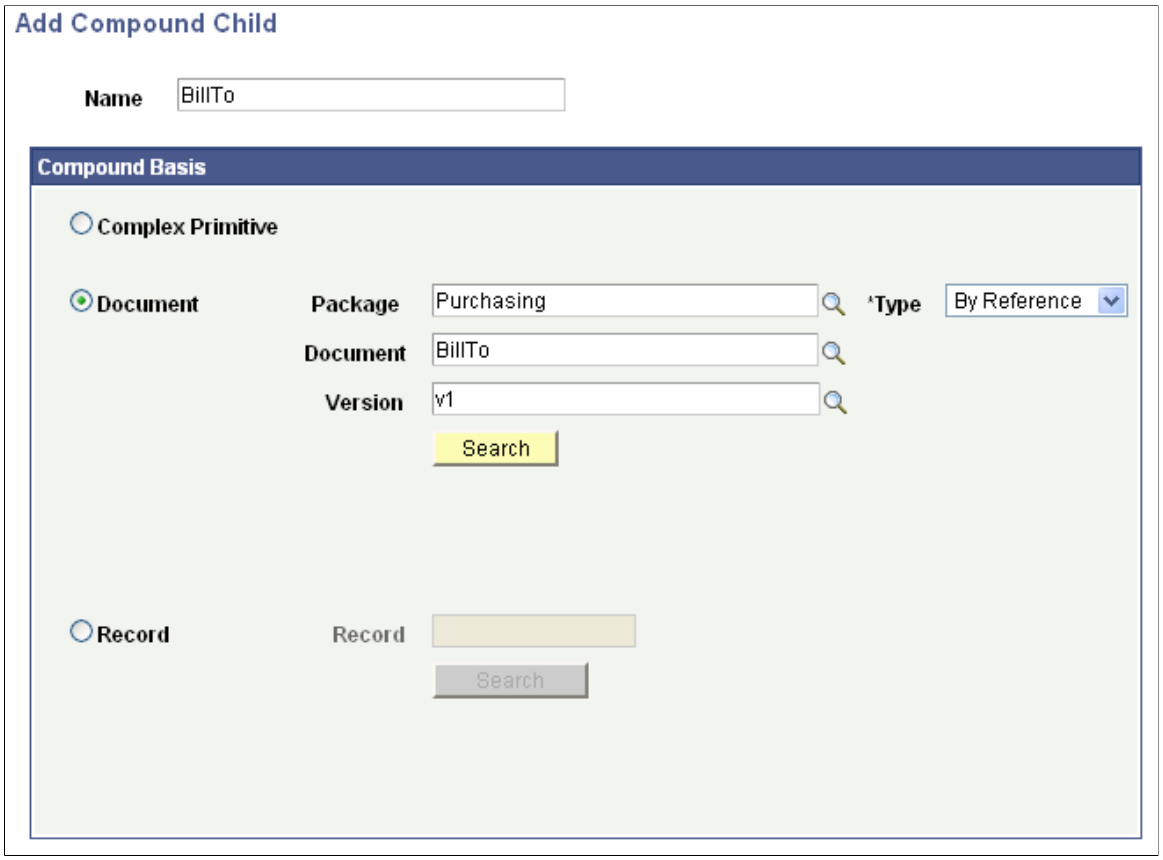
See Managing Primitive Elements, Managing Compound Elements, Managing Collection Elements.

Defining Documents as Compound Element Data Types

After you create and name a document as the data type for a compound element, you specify the document package, document name, and document version on the Add Compound Child or Add Compound Peer page (IB_LOGICALCOMP_SEC).

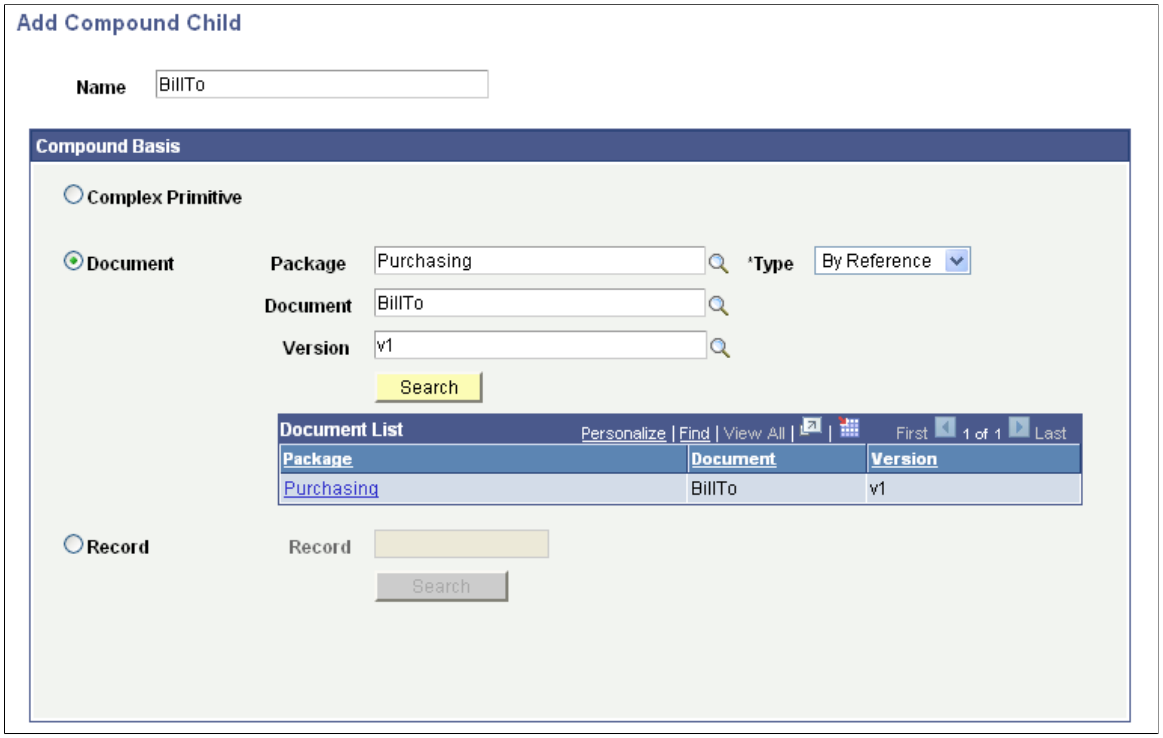
When you specify a document as the compound element data type, you can select to use a reference of the document or a copy of the document. Using a reference of the document means that if the document defined as the compound element type is modified, then the changes will be rolled into that document. Using a copy of the document means that the document you define as the compound element type will remain the same, even if the original document is modified in the future.

This example shows specifying the document to add as a compound element data type.



After you enter the document details, click the Search button to select the document to add as a compound element.

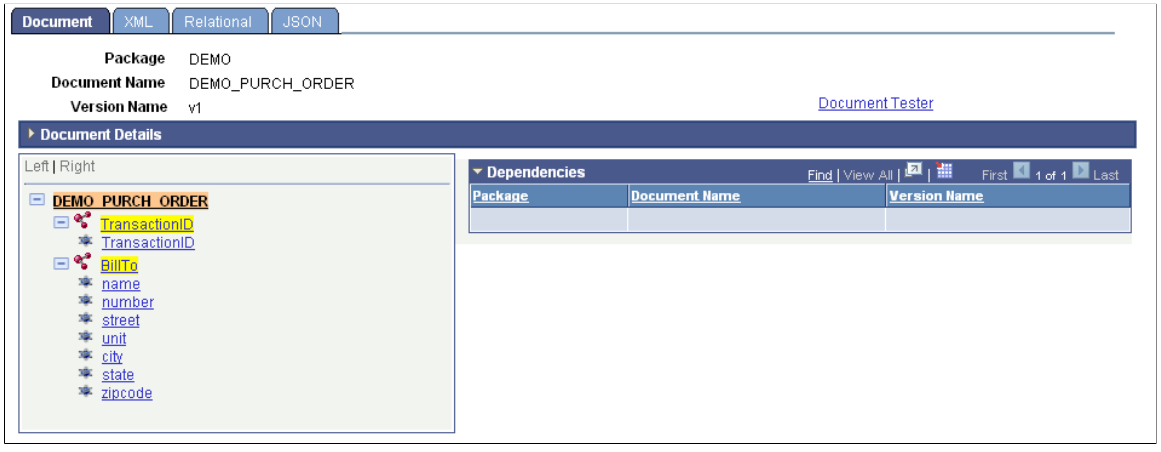
The following example shows the search results from the parameters entered in the previous example.



Note: To select from all documents in the database, do not enter any document information and click the Search button.

After you search for and select a document, the Document Builder - Documents page appears, and the document appears as a compound element in the document tree.

The following example shows the *BillTo* document added as a compound element to the *DEMO_PURCH_ORDER* document:



After you create a document as a compound element data type, the document element information appears on the right side of the page. A **Go to Child Document** link also appears. When you click the **Go to Child Document** link, the document definition for the compound element appears in a new window.

To define a document as a compound element data type:

1. Create and name a document as a data type for a compound element.

   See Adding Compound Elements.

2. In the **Type** drop-down list box, select an option:

   - *By Reference.* (Default.)

     Use a reference of the document.

   - *Copy.*

     Use a copy of the document.

3. Specify the document:

   a. In the **Package** field, enter the document package name or click the **Lookup** button to search for one.

   b. In the **Document** field, enter the document name or click the **Lookup** button to search for one.

   c. In the **Version** field, enter the document name or click the **Lookup** button to search for one.

   d. Click the **Search** button.

      The search results appear in the **Document List** grid.

   e. Select a document from the list.

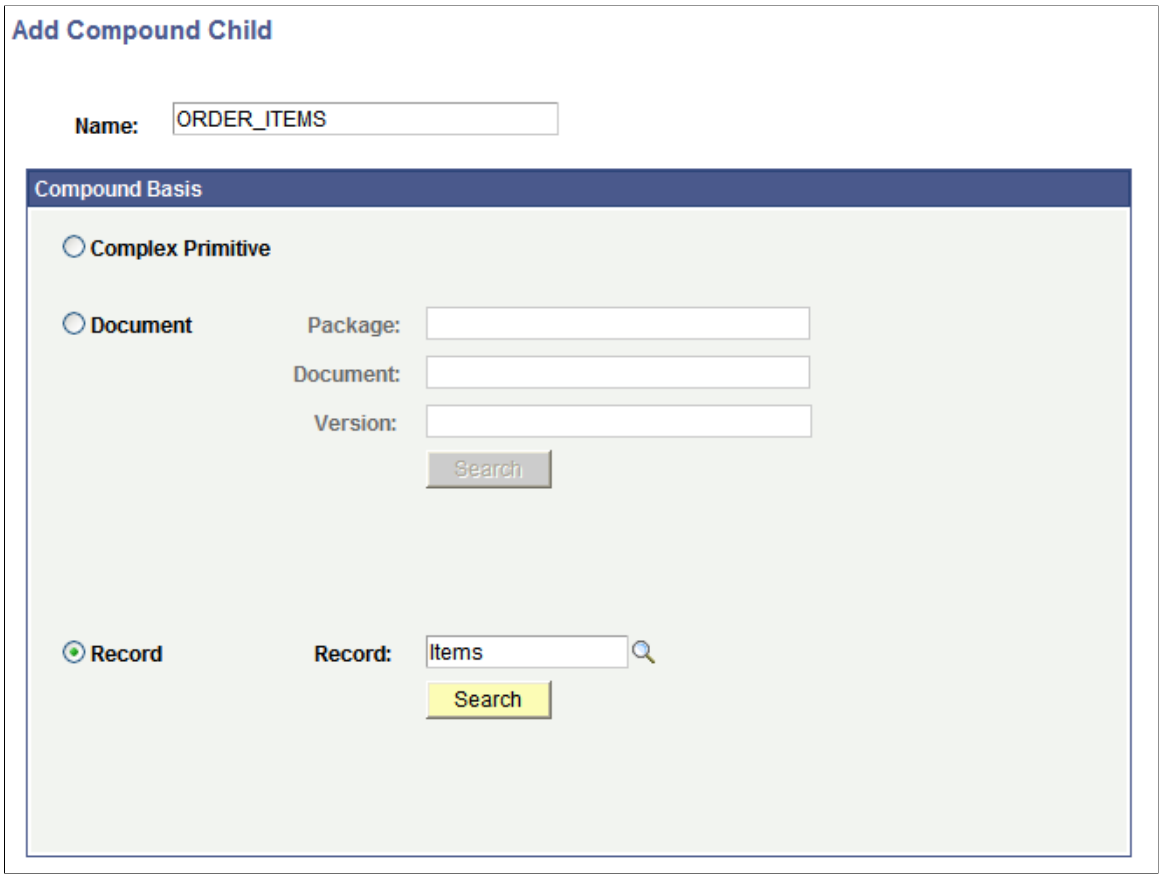
      The Document Builder - Documents page appears, and the document you specified appears in the document tree.

4. Click the **Save** button.

Defining PeopleSoft Records as Compound Element Data Types

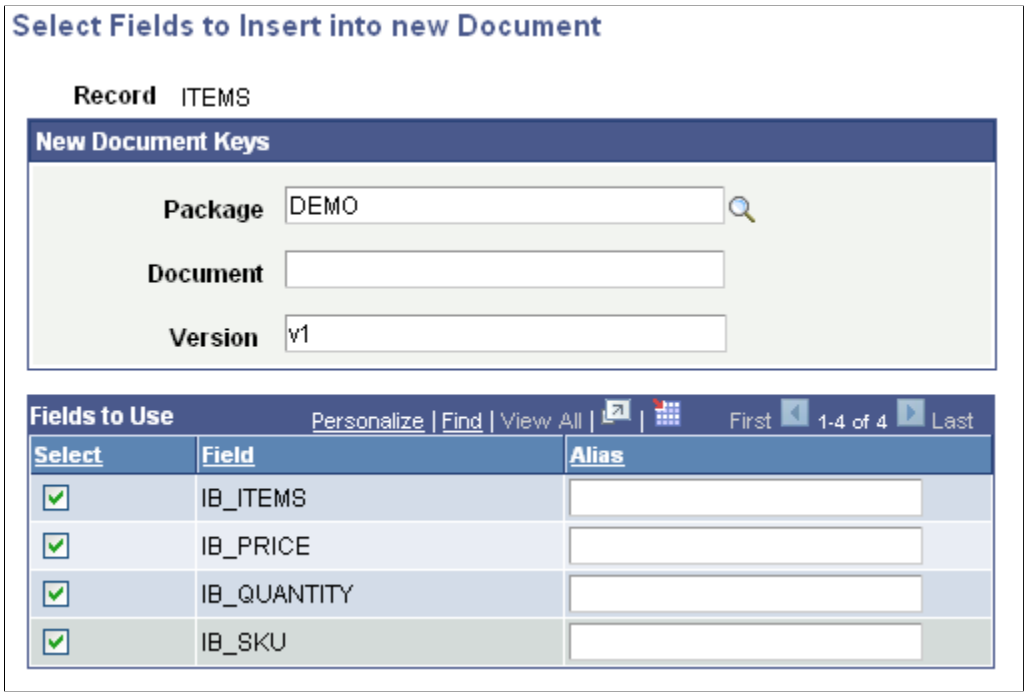
After you name a compound element and choose to create a PeopleSoft record data type, you must search for and select the record. As with other compound data types, you use the Add Compound Child or Add Compound Peer page (IB_LOGICALCOMP_SEC), depending on whether you are creating a compound child or compound peer element.

The following example shows the Add Compound Child page when searching for a PeopleSoft record to add as a compound child.



After you enter search criteria, the results appear in the **Records** grid. When you select a record from the grid, the Select Fields to Insert into New Document page (IB_LOGICALFLD_SEC) appears. Use the page to select the fields to include in the document. The fields become primitive elements in the document, and the record fields properties, such as data type and length, become the default primitive element properties in the document.

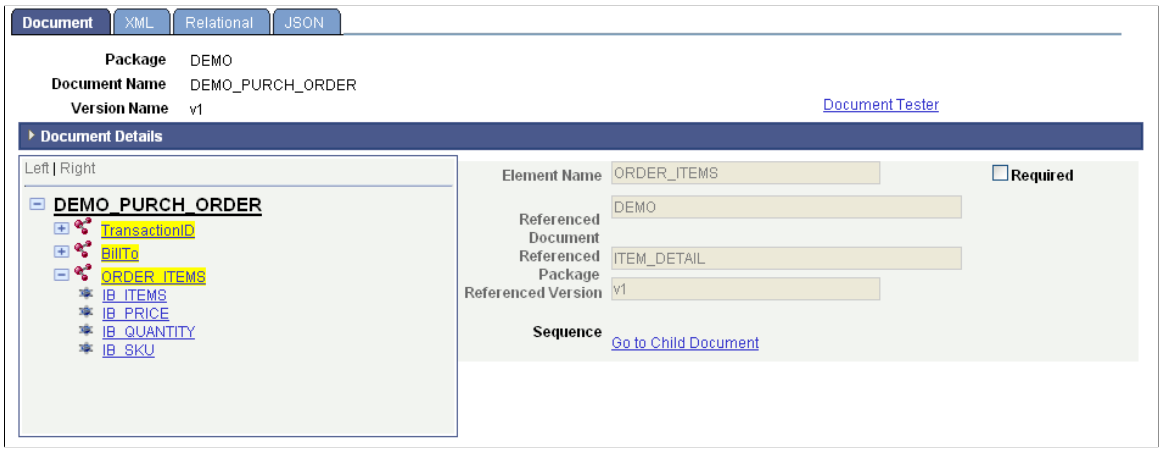
This example shows the available fields from the *ITEMS* record that you can insert and use in *ORDER_ITEMS* compound element.



When you save the information, the record and its fields become a compound document element. In the previous example, you see that the system reads the package name and version from the document definition and supplies the document name in the **Document** field.

After you provide the document name and click the **OK** button, the new compound document element that you created from a PeopleSoft record appears in the document tree on the Document Builder - Document page.

The following example shows the compound element, *ORDER_ITEMS,* appearing in the document tree.



The previous example shows that the system converted the PeopleSoft *ITEMS* record and its fields into a document called *ORDER_ITEMS,* and that the document is a compound child element of the *DEMO_PURCH_ORDER* document.

Click the Go to Child Document link in the properties section of the page to view and modify the *ORDER_ITEMS* document definition.

To define a PeopleSoft record as a compound element:

1. Access the Document Builder–Document page (**PeopleTools** > **Documents** > **Document Builder**).
2. On the document tree, select the element to which to add the compound element and on the Action Toolbar, click the **Add Compound Child** or **Add Compound Peer** button as appropriate.

   The Add Compound Child or Add Compound Peer page appears, depending on the toolbar button selected.
3. Select the PeopleSoft record.
   a. Click the **Record** radio button.
   b. In the **Record** field, enter the record name or use the **Lookup** button to search for one and click the **Search** button.

      A Records list appears.
   c. From the Records list, click the record name to use.

      The Select Fields to Insert into new Document page appears.
4. Select the fields to insert into the new document.
   a. In the New Document Keys section, enter the document package, name and version for the new document.
   b. In the Fields to Use section, use the **Select** box to select or deselect the fields to include in the new document.
   c. In the **Alias** column, enter field alias names to use.
5. Click the **OK** button.

Managing Collection Elements

These topics provide an overview of managing collection elements and discuss how to:

- Add collection elements.
- Define collection element properties.

Understanding Managing Collection Elements

Only a collection item that is a primitive or compound data type can be added to a collection element. A collection can have unlimited collections as siblings.

Adding Collection Elements

Use the Add Collection Child or the Add Collection Peer page, depending on whether you want to create a collection child or collection peer element. The Add Collection Child page and the Add Collection Peer page share the same object page name: IB_LOGICALCOLL_SEC.

You access the Add Collection Child and Add Collection Peer pages using the action toolbar on the Document Builder - Documents page.

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

To add a collection element:

1. Access the Document Builder - Documents page (**PeopleTools** > **Documents** > **Document Builder**).
2. In the document tree, click the element to which to add a collection child or collection peer element.
3. On the action toolbar, do one of the following:
   - Click the**Add Collection Child** button to add a collection child element.
   - Click the **Add Collection Peer** button to add a collection peer element.

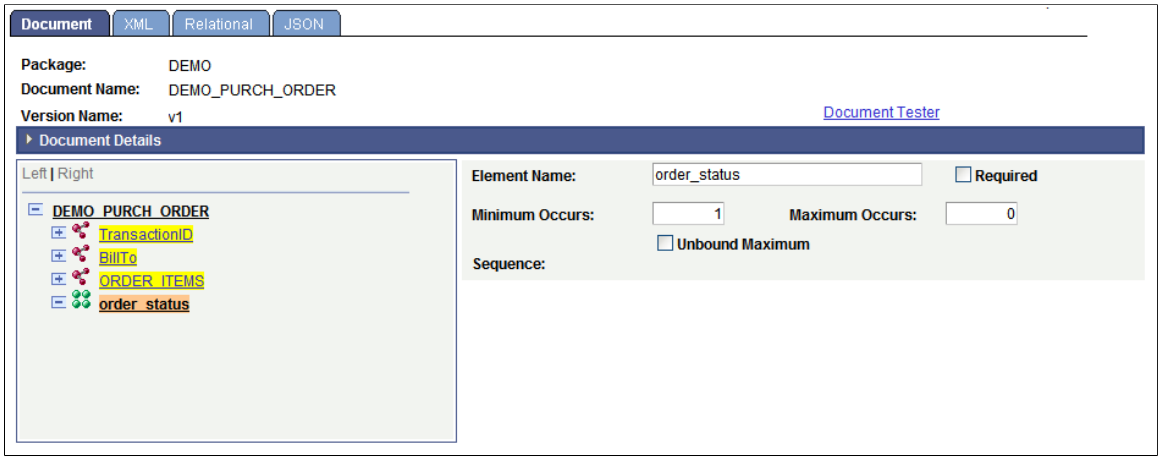
   The Add Collection Child or Add Collection Peer page appears, depending on the element you added.
4. In the **Element Name** field, enter a name for the collection element.
5. Click the **OK** button.

   The Document Builder - Documents page appears, and the new collection element appears in the document tree. Note that the collection element icon that appears on the toolbar also appears to the left of the collection element in the document tree.

Defining Collection Element Properties

After you add a collection child or collection peer element to a document, the collection element properties appear on the right side of the Document Builder - Documents page.

The following example shows the properties for a collection child element called *order_status*.



You can define the following collection element properties:

Field or Control	***Description***
Element Name	Name of the element as it will appear in the XML document.
Required	Select the check box to make the element required.
Min Occurrence (minimum occurrence)	The minimum number of instances when populating the document definition with data. As an example, entering a *0* (zero) means that you do not have to populate the collection. Enter the minimum number of occurrences of the collection element in the document. The default value is *1*.
Max Occurs (maximum occurrence)	This field appears only if the **Unbound Maximum** field is cleared. Enter the maximum number of occurrences of the collection element in the document.
Unbound Maximum	Select the check box to include an unlimited number of occurrences of the collection element in the document. By default, this option is selected.
Sequence	The system assigns this read-only field. The sequence number is used for indexing with the PeopleCode API for documents.

Chapter 7

Managing Formatted Documents

Understanding Formatted Documents

After you have built the logical format of a document by adding elements and defining element properties, you can build the physical document. The physical document specifies the concrete details of a document, such as database tables, XML tag names, and so on.

Managing XML-Formatted Documents

This section discusses how to:

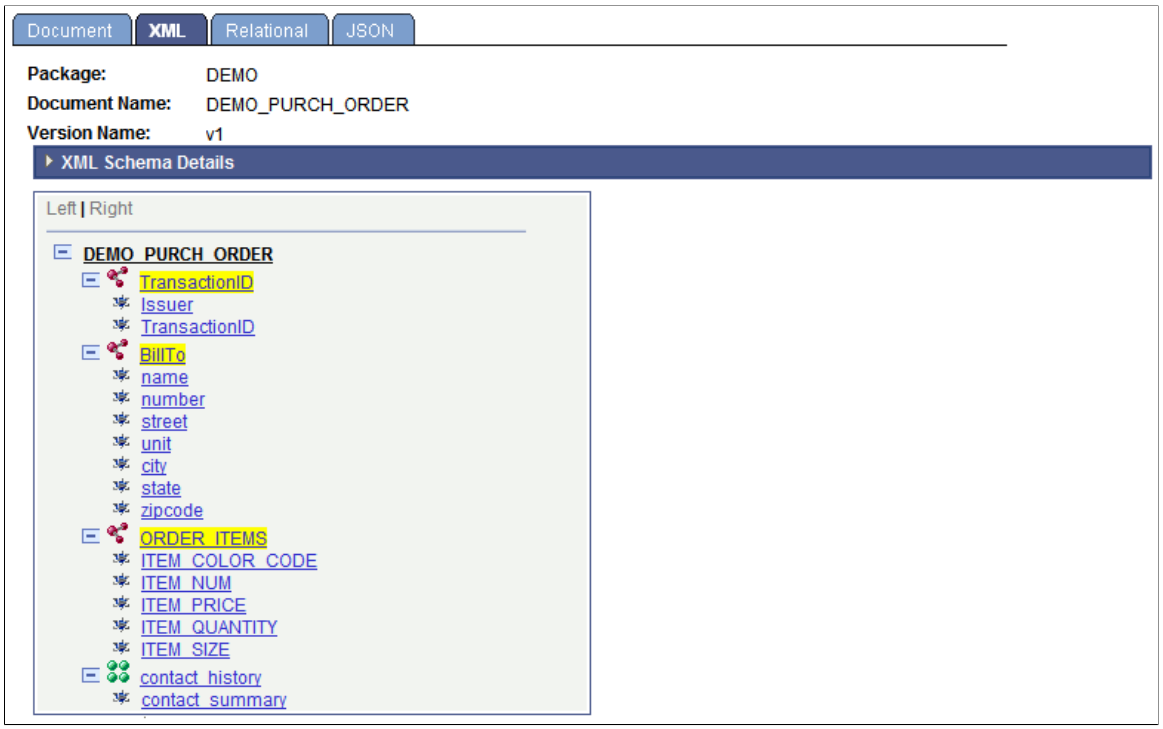
- Define XML schema details.
- Define element details.
- Validate XML schemas.

Defining XML Schema Details

Use the Document Builder - XML page (IB_XMLSCHEMA) to define XML schema details for a document. This page enables you to define root tag names and namespace prefixes, include or exclude the schema reference in the generated XML, filter blank elements, and more.

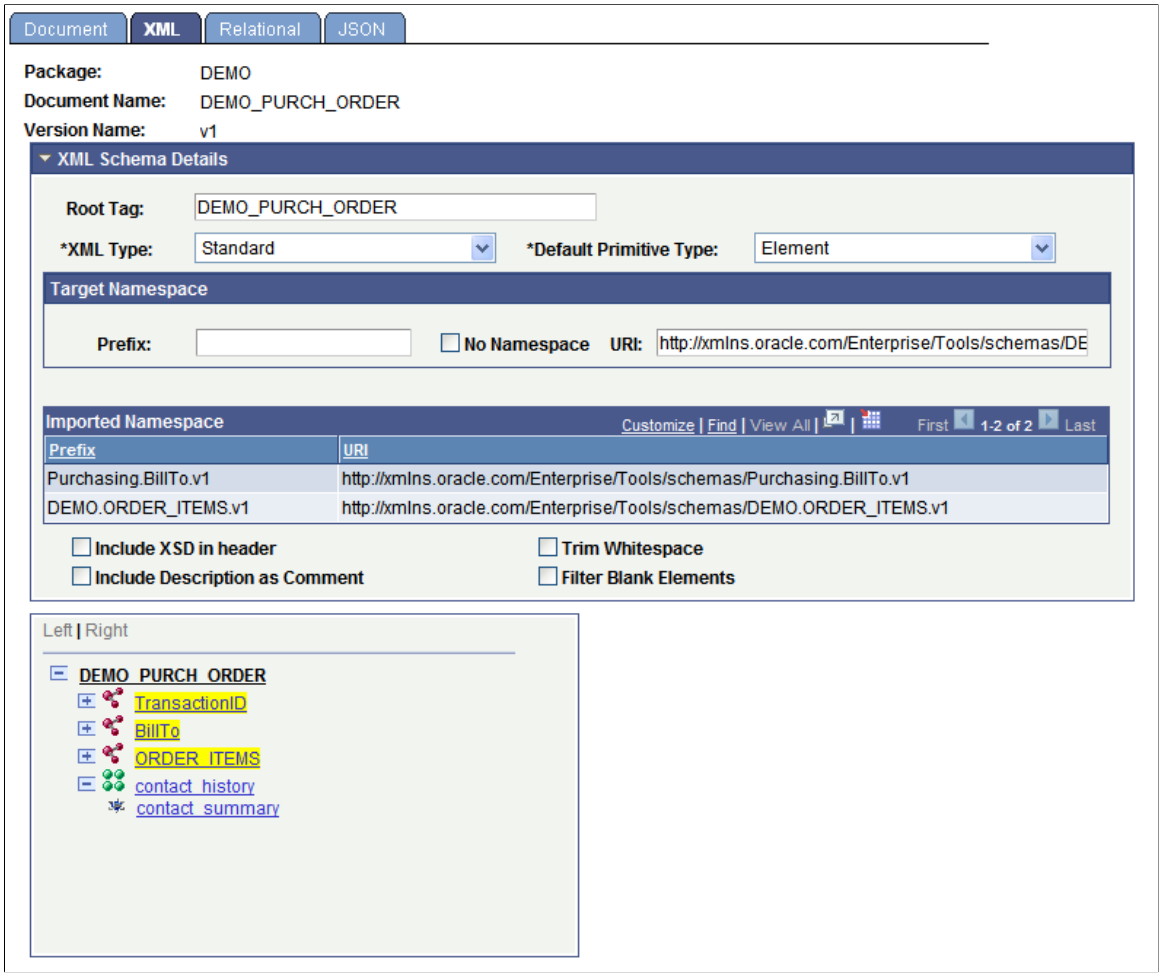
To access the page, select *PeopleTools, Documents, Document Builder* and click the *XML* tab.

This example illustrates the fields and controls on the Document Builder - XML page. You can find definitions for the fields and controls later on this page.



When you initially open the Document Builder - XML page, the XML Schema Details section is collapsed. Click the arrow icon next to the XML Schema Details label to expand the section.

This example shows the XML Schema Details section of the Document Builder – XML page expanded.



The following page elements appear in the XML Schema Details section:

Field or Control	*Description*
Root Tag	(Optional) Enter the root tag as it is to appear in the XML document schema. By default, the system populates the field with the document name. By default, the system populates the field with the only the first 30 characters of the document name. However, the full document name, up to the document name character limit of 100, appears in the generated XML document schema. The exception is if you modify the **Root Tag** field and overwrite the default with a new value; if you choose to do so, that value is used as the root tag element in generated schema. See Understanding Naming Document Definitions and XML Root Tag Names.

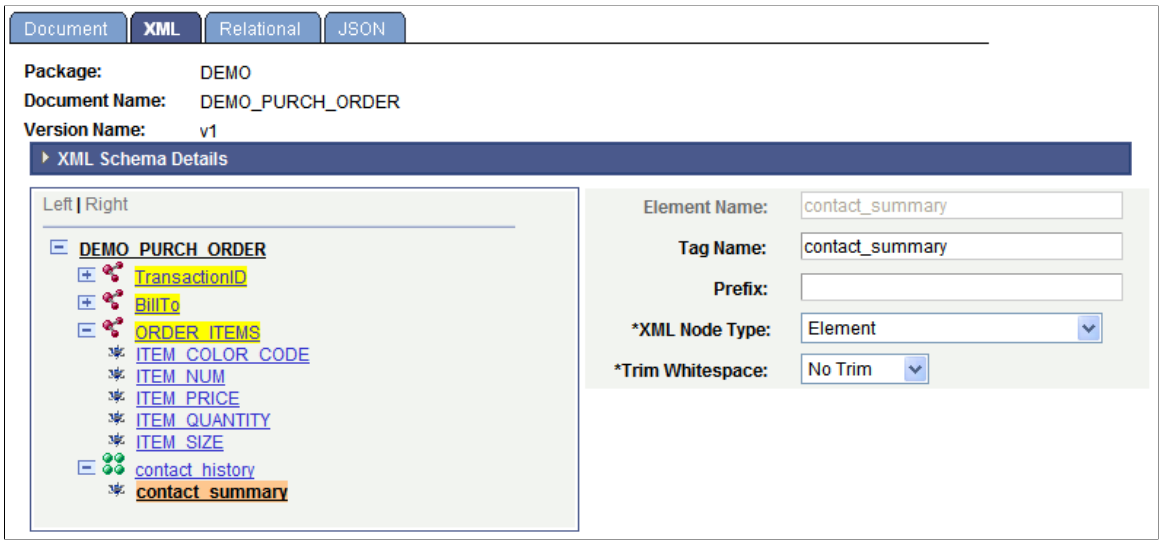
Field or Control	*Description*
XML Type	Select the XML type from the drop-down list. Currently, *Standard* is the only available XML type and is the default value.
Default Primitive Type	Select a value from the drop-down list. The options are: • *Element:* The primitive data types appear as elements in the XML schema. This value is the default. • *Element and CDATA Section:* The primitive data types appear within a CDATA section in the XML schema.
Prefix	(Optional) Enter a target namespace prefix.
No Namespace	(Optional) Select the check box to exclude the document target namespace from the XML schema. You can still use a prefix with this option enabled. Use this option when you want the document to be used as a child document. The document will use the target namespace of the parent. This option enables you to create libraries with schemas that are namespace transparent.
URI	Enter a target namespace. By default, the system uses the target namespace defined on the Integration Broker Service Configuration page.
Imported Namespace Section	This section displays the namespace prefix and URI of imported document namespaces. When you add a compound element that is a document data type to a document, the namespace prefix and URI are imported into the document definition.
Include XSD in header	(Optional) Select the check box to include the schema reference in the generated XML.
Trim Whitespace	(Optional) Select the check box to trim leading and trailing whitespace in the generated XML.
Include Description as Comment	(Optional) Select the check box to include descriptions in the generated XML schema.
Filter Blank Elements	(Optional) Select the check box to exclude blank elements in the generated XML. The IsChanged property must be false for primitive PeopleCode objects.

Defining Element Details

The Document Builder - XML page features an element details section where you can define element tag names, prefixes, and other details. To access the page, select *PeopleTools, Documents, Document Builder* and click the *XML* tab. The area you use to define element details appears under the XML Schema Details section.

To display the section, select an element. The details that you can define and manage appear on the right side of the page.

This example illustrates the fields and controls on the Document Builder - XML page. You can find definitions for the fields and controls later on this page.



This example shows the element details section that appears when the *contact_summary* element in the document tree of the *DEMO_PURCH_ORDER* document is selected. Note that the XML Schema Details section is collapsed:

The following page elements appear in the XML Schema Details section:

Field or Control	***Description***
Element Name	This read-only field displays the name of the selected element.
Tag Name	(Optional) Enter the tag name to appear in the generated XML schema. By default, the system populates the field with the element name.
Prefix	(Optional) Enter a prefix for the tag name.

Field or Control	*Description*
XML Node Type	From the drop-down list, select the XML node type for the element in the generated XML. The options are: • *Attribute:* The element will appear as an element attribute in the generated XML. • *Element:* The element will appear as an element node in the generated XML. This value is the default. • *Element and CDATA Section:* The element will appear as an element within a CDATA section in the XML schema.
Trim Whitespace	Select whether to trim the leading and trailing whitespace in the generated XML. The options are: • *No Trim:* Leading and trailing whitespace is not trimmed in the generated XML. This value is the default. • *Trim:* Leading and trailing whitespace is trimmed in the generated XML.

Validating XML Schemas

The Document Builder - XML page enables you to validate an XML schema for a document. To access the page, select **PeopleTools** > **Documents** > **Document Builder** and click the **XML** tab. To validate an XML schema for a document, click the **Validate** button at the bottom of the page. The system displays the validation results in a separate window.

Managing Relational-Formatted Documents

This section provides an overview of managing relational-formatted documents and discusses how to:

- Map PeopleSoft records to documents.
- Map document elements to PeopleSoft record fields.

Understanding Managing Relational-Formatted Documents

You can use the Document Builder to map PeopleSoft records and fields to documents. Doing so enables you to populate a rowset with a document.

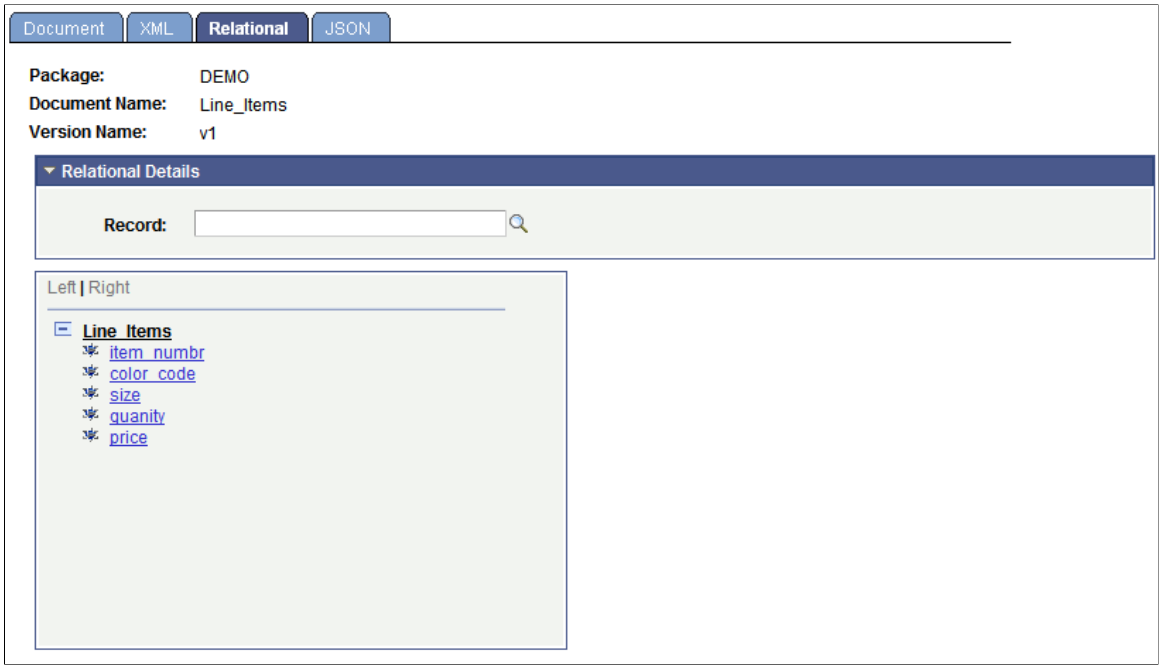
Mapping PeopleSoft Records to Documents

Use the Document Builder - Relational page (IB_RELATSCHEMA) to map a PeopleSoft record to a document. To access the page, select **PeopleTools** > **Documents** > **Document** > **Document Builder** and click the *Relational* tab.

This example illustrates the fields and controls on the Document Builder - Relational page. You can find definitions for the fields and controls later on this page.

When you first access the page, the Relational Details section is collapsed. The Relational Details section is where you select the PeopleSoft record to map to the document. Click the arrow icon next to the Relational Details label to expand the section.

This example illustrates the Document Builder – Relational page with the Relational Details section expanded.



To map a PeopleSoft record to a document:

1. Access the Document Builder - Relational page (**PeopleTools** > **Documents** > **Document Builder** and click the Relational tab).
2. Expand the Relational Details section.
3. In the **Record** field, enter the name of the PeopleSoft record to map to the document or use the **Lookup** button to select one.

Before you save the record, you should map the record fields to the document elements. If you attempt to save the definition before mapping the document elements to the fields in the record you specified, the system displays an error message. You can save the record without defining all of the field maps and return to the definition at a later time to specify the remaining maps. However, the document will fail validation until you define all maps.

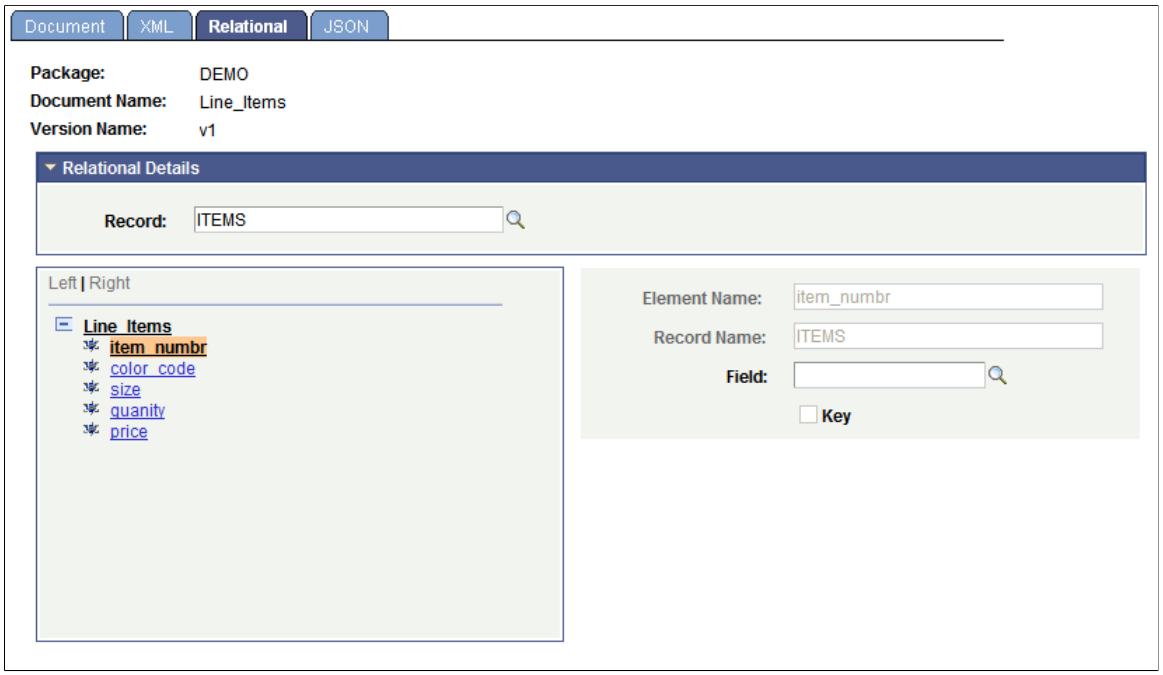
The procedure for mapping elements to record fields is described in the next section.

Mapping Document Elements to PeopleSoft Record Fields

After you map a PeopleSoft record to a document, you must map the document elements to the record fields.

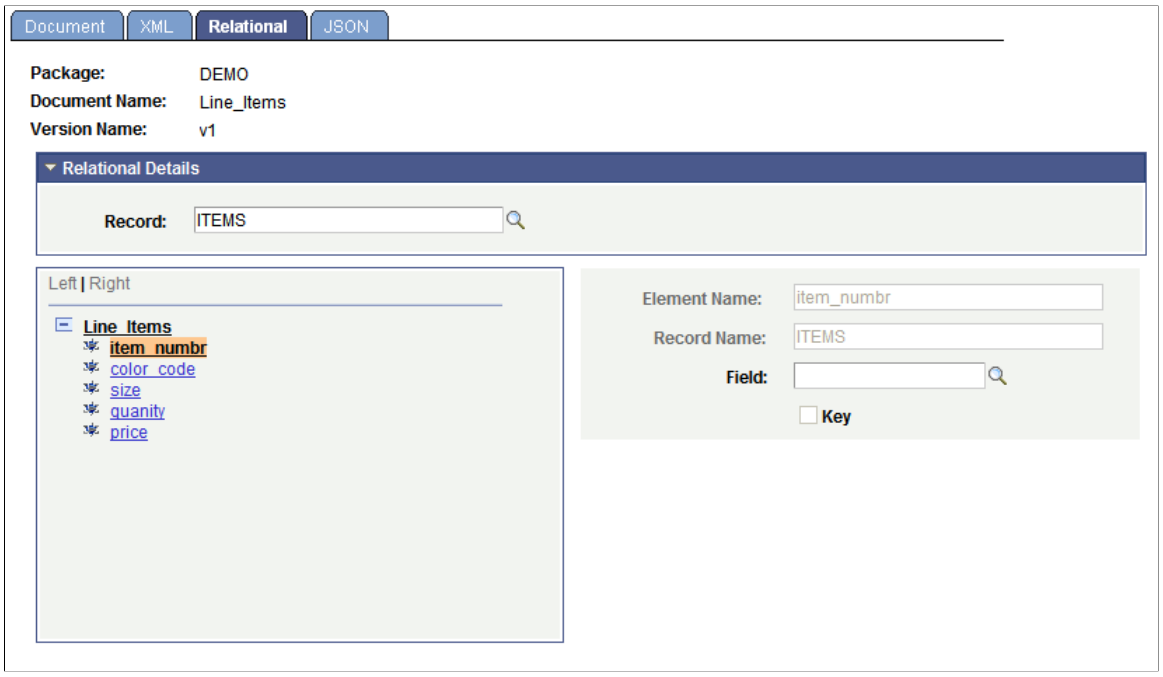
After you map a record to a document, when you select an element in the document tree, a properties section displays to the right of the document tree, where you can map the element to a record field.

The following example shows a record called *ITEMS* that is mapped to the document *Line_Items*. In the document tree, the element *item_numbr* is selected, and the section to map a record field to the element appears on the right side of the page.



In the properties section, you can enter a field name or use the **Lookup** button to select from all fields in the record. You also have the option of specifying the field as a key field.

This example shows that the record field *ITEM_NUM* has been selected to be mapped to the *item_numbr* element.



To map document elements to PeopleSoft record fields

1. In the document tree, click the name of an element.

   The relational elements properties section appears on the right side of the page.

2. In the **Field** field, enter the name of the field to map to the element, or use the *Lookup* button to select one.

3. Select the **Key** check box if the field to define the field as a key field.

4. Repeat steps 1 through 3 to map each element in the document tree.

5. Click the **Save** button.

Managing JSON-Formatted Documents

If a JSON structure needs to be defined with null, for example to send to a 3rd party, consider not using Documents instead use the JSON Object to create your JSON string.

When Documents is used to parse JSON where the values are defined as null, the parser will simply use the default values based on the primitive type. For example a string would be “ ” and an integer would be 0.

Define Root Labels for JSON Documents

When you create a JSON document, by default the system uses the root element label from the logical document as the JSON root label. You can use the default root label or define a different root label for the JSON document.

Use the Document Builder – JSON page to define a root element label for a JSON document. To access the page, select **PeopleTools** > **Documents** > **Document Builder** and click the JSON tab.

This example illustrates the Document Builder – JSON page. Use the JSON Details section to define a JSON root label.

To define a root label for a JSON document:

1. Access the Document Builder – JSON page (**PeopleTools** > **Documents** > **Document Builder** and click the JSON tab).
2. In the JSON Details section, click the arrow icon to expand the section.
3. In the **JSON Label** field enter a label for the JSON document.
4. Click the **Save** button.

Note that if the **JSON Label** field is left blank, the system will default the field value to the root label of the logical document when you save the document.

Defining Tag Names for JSON Document Elements

By default for any element, the element name defined in the logical document is used as the JSON tag name for the element. However, you are able to define any JSON tag name you wish for an element.

Use the Document Builder – JSON page to define JSON tag names. To access the page, select **PeopleTools** > **Documents** > **Document Builder** and click the JSON tab.

When you access the Document Builder – JSON page and select an element in the document tree, on the right side of the page the name of the element as defined in the logical document appears in a read-only **Element Name** field for your reference. A **JSON Tag Name** field appears on the right side of the page

as well. The field is populated with the default name of the element from the logical document. You can define a different value for the field for use when JSON data is generated or parsed.

In the previous example, the *name* element in the *BillTo* document is highlighted in the document tree. On the right side of the page the default element name, **name,** appears in the read-only **Element Name** field. The **JSON Tag Name** field shows that a value *IB_NAME* has been defined as the JSON tag name for the *name* element.

To define tag names for JSON document elements:

1. Access the Document Builder – JSON page (**PeopleTools** > **Documents** > **Document Builder** and click the JSON tab).
2. Click an element in the document tree.

   The **Element Name** field and the **JSON Tag Name** field appear on the right side of the page.
3. In the **JSON Tag Name** field, enter a name.
4. Click the **Save** button.

Omitting the JSON Label in Generated JSON

Use the Hide Parent Object Label box in the JSON Details section of the Document Builder – JSON page to omit the JSON label (object name of the parent document) in the generated JSON output.

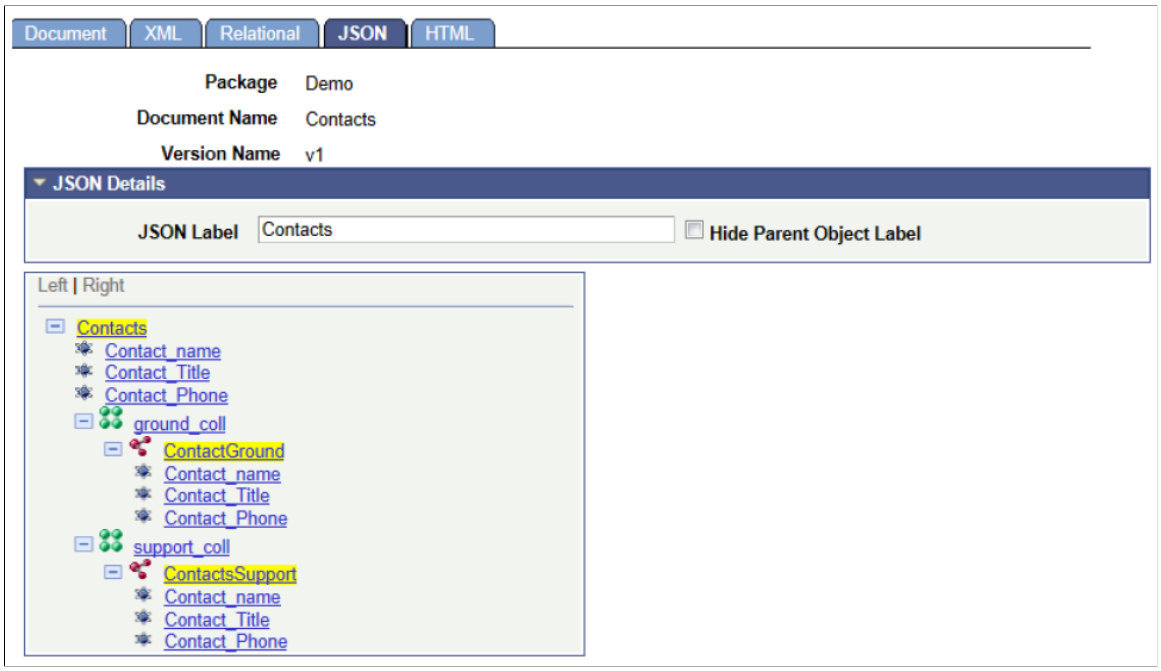
Doing so provides more flexibility in creating third–party JSON strings.

The Hide Parent Object Label box is enabled only when a document is created using the Integration Broker Consume Web Service wizard.

Note: If the check box is not selected the JSON string to parse must have the parent JSON label present in the string and in the correct format.

The following examples show generated JSON for a document when the Hide Parent Object Label option is not used and when it is used.

The examples in this section are based on the following PeopleSoft document.



This example illustrates the generated JSON when the Hide Parent Object Label is not used:

```
{"Contacts": {
"Contact_name": "Robby Naish","Contact_Title": "Flight Test Director","Contact_Phon⇒

e": "x4354",
"ContactGround":
[
{"Contact_name": "Rob Tock","Contact_Title": "Ground Supervisor","Contact_phone": "⇒

x6577"},{
"Contact_name": "David Bourn","Contact_Title": "Ground Support","Contact_phone": "x⇒

6577"}
],"ContactsSupport": [
{"Contact_name": "Scott McDermott","Contact_Title": "Flight Support","Contact_phone⇒

": "x6533"}
]}
}
```

This example illustrates the generated JSON when the Hide Parent Object Label is used:

```
{
"Contact_name": "Robby Naish","Contact_Title": "Flight Test Director","Contact_Phon⇒

e": "x4354",
"ContactGround":
[
{"Contact_name": "Rob Tock","Contact_Title": "Ground Supervisor","Contact_phone": "⇒

x6577"},{
"Contact_name": "David Bourn","Contact_Title": "Ground Support","Contact_phone": "x⇒

6577"}
],"ContactsSupport": [
{"Contact_name": "Scott McDermott","Contact_Title": "Flight Support","Contact_phone⇒

": "x6533"}
```

```
        ] }
```

Chapter 8

Populating and Retrieving Document Data

Understanding Populating and Retrieving Document Data

PeopleSoft provides a Document class that contains built-in functions and methods that enable you to populate and retrieve document data.

This topic provides PeopleCode examples for your reference.

Note: The Document class is described in detail in the product documentation for *PeopleTools 8.55: PeopleCode API Reference*

The Document class provides PeopleCode for populating documents. It also provides PeopleCode for reading data from primitive, compound, and collection elements.

Document PeopleCode specific to Integration Broker messages is described in the product documentation for *PeopleTools 8.55: PeopleSoft Integration Broker*

Related Links

Populating Document Data
Retrieving Document Data
"Understanding the Document Classes" (PeopleCode API Reference)

Populating Document Data

The following pseudo code is an example of populating a document using the CreateDocumentKey built-in function.

This function enables you to pass a document key into the CreateDocument built-in function, which then gives you the document.

```
Local Document &DOC;
Local DocumentKey &DOCKEY;
Local Primitive &PRIM;

/* Populating Document Object */
&DOCKEY = CreateDocumentKey("Purchasing", "PurchaseOrder", "v1");
&DOC = CreateDocument(&DOCKEY);

&COM = &DOC.DocumentElement;

&COM.GetPropertyByName("Language_Code").Value = "ENG";

&COM_TRID = &COM.GetPropertyByName("TransactionId");
&COM_TRID.GetPropertyByName("issuer").Value = "PSFT";
&COM_TRID.GetPropertyByName("TransactionId").Value = "26435383";

&COM_BILLTO = &COM.GetPropertyByName("BillTo");
```

```
&COM_BILLTO.GetPropertyByName("name").Value = "Robby Naish";
&COM_BILLTO.GetPropertyByName("number").Value = "234";
&COM_BILLTO.GetPropertyByIndex(3).Value = "Alpine";
&COM_BILLTO.GetPropertyByIndex(4).Value = "4";
&COM_BILLTO.GetPropertyByName("city").Value = "Hood River";

&PRIM = &COM_BILLTO.GetPropertyByName("state");
&PRIM.Value = "Oregon";
&COM_BILLTO.GetPropertyByName("zipcode");
&PRIM.Value = "97031";

&COM_SHIPTO = &COM.GetPropertyByName("ShipTo");
&COM_SHIPTO.GetPropertyByName("name").Value = "Naish Sails";
&COM_SHIPTO.GetPropertyByName("number").Value = "123";
&COM_SHIPTO.GetPropertyByIndex(3).Value = "High Wind";
&COM_SHIPTO.GetPropertyByIndex(4).Value = "1";
&COM_SHIPTO.GetPropertyByName("city").Value = "Hood River";
&COM_SHIPTO.GetPropertyByName("state").Value = "Oregon";
&COM_SHIPTO.GetPropertyByName("zipcode").Value = "97031";

&COll_ITEMS = &COM.GetPropertyByName("item_collection");

&COM_ITEM = &COll_ITEMS.CreateItem();
&COM_ITEM.GetPropertyByName("items").Value = "4.2 Sail";
&COM_ITEM.GetPropertyByName("sku").Value = "s12322";
&COM_ITEM.GetPropertyByName("price").Value = "450";
&COM_ITEM.GetPropertyByName("quantity").Value = "2";
&nRet = &COll_ITEMS.AppendItem(&COM_ITEM);

&COM_ITEM = &COll_ITEMS.CreateItem();
&COM_ITEM.GetPropertyByIndex(1).Value = "Mast";
&COM_ITEM.GetPropertyByName(2).Value = "m485765";
&COM_ITEM.GetPropertyByName(3).Value = "550";
&COM_ITEM.GetPropertyByName(4).Value = "3";
&nRet = &COll_ITEMS.AppendItem(&COM_ITEM);

&COM_ITEM = &COll_ITEMS.CreateItem();
&COM_ITEM.GetPropertyByName("items").Value = "Boom";
&COM_ITEM.GetPropertyByName("sku").Value = "b9754";
&COM_ITEM.GetPropertyByName("price").Value = "375";
&COM_ITEM.GetPropertyByName("quantity").Value = "2";
&nRet = &COll_ITEMS.AppendItem(&COM_ITEM);
```

Retrieving Document Data

The following pseudo code is an example of retrieving data out of a compound element using the GetPropertyByName and GetPropertyByIndex methods:

```
&COM = &DOC.DocumentElement;

&LNG_Code = &COM.GetPropertyByName("Language_Code").Value;

&COM_TRID = &COM.GetPropertyByName("TransactionId");
&data = &COM_TRID.GetPropertyByName("issuer").Value;
&data = &COM_TRID.GetPropertyByName("TransactionId").Value;

&COM_BILLTO = &COM.GetPropertyByName("BillTo");
&data = &COM_BILLTO.GetPropertyByName("name").Value;
&data = &COM_BILLTO.GetPropertyByName("number").Value;
&data = &COM_BILLTO.GetPropertyByIndex(3).Value;
&data = &COM_BILLTO.GetPropertyByIndex(4).Value;
&data = &COM_BILLTO.GetPropertyByName("city").Value;
&data = &COM_BILLTO.GetPropertyByName("state").Value;
&data = &COM_BILLTO.GetPropertyByName("zipcode").Value;

&COM_SHIPTO = &COM.GetPropertyByName("ShipTo");
```



```
&PRIM = &COM_SHIPTO.GetPropertyByName("name");
&data = &PRIM.Value;
&data = &COM_SHIPTO.GetPropertyByName("number");
&data = &PRIM.Value;
&data = &COM_SHIPTO.GetPropertyByIndex(3).Value;
&data = &COM_SHIPTO.GetPropertyByIndex(4).Value;
&data = &COM_SHIPTO.GetPropertyByName("city").Value;
&data = &COM_SHIPTO.GetPropertyByName("state").Value;
&data = &COM_SHIPTO.GetPropertyByName("zipcode").Value;

&COll_ITEMS = &COM.GetPropertyByName("item_collection");

For &i = 1 To &COll_ITEMS.GetCount

   &COM_ITEM = &COll_ITEMS.GetItem(&i);
   &data = &COM_ITEM.GetPropertyByName("items").Value;
   &data = &COM_ITEM.GetPropertyByName("sku").Value;
   &data = &COM_ITEM.GetPropertyByName("price").Value;
   &data = &COM_ITEM.GetPropertyByName("quantity").Value;

End-For;
```


Chapter 9

Creating Documents from Schema

Understanding Creating Documents from Schema

PeopleSoft provides a Create Document from Schema utility that enables you to create a document based on an XML schema that you import using an XSD URL or from a file.

When you create a document from schema, you can create the document in an existing package or specify a new package.

Note: You can process the schema to build a document. However, due to the complexities associated with schemas and the variations on how they can be created, the document generated might not be correct. At this point, the developer need to determine if the document properly reflects the schema and then modify the document accordingly.

Accessing the Create Document from the Schema Utility

The Create Document from Schema utility page (IB_DOCUMENT_CRSCHM) is located in the IB_DOCUMENT_CRSCHM component.

To access the Create Document from Schema Utility, select **PeopleTools** > **Documents** > **Document Utilities** > **Create Document from Schema.**

This example shows the Create Document from Schema page.

Reading Schemas

To read a schema:

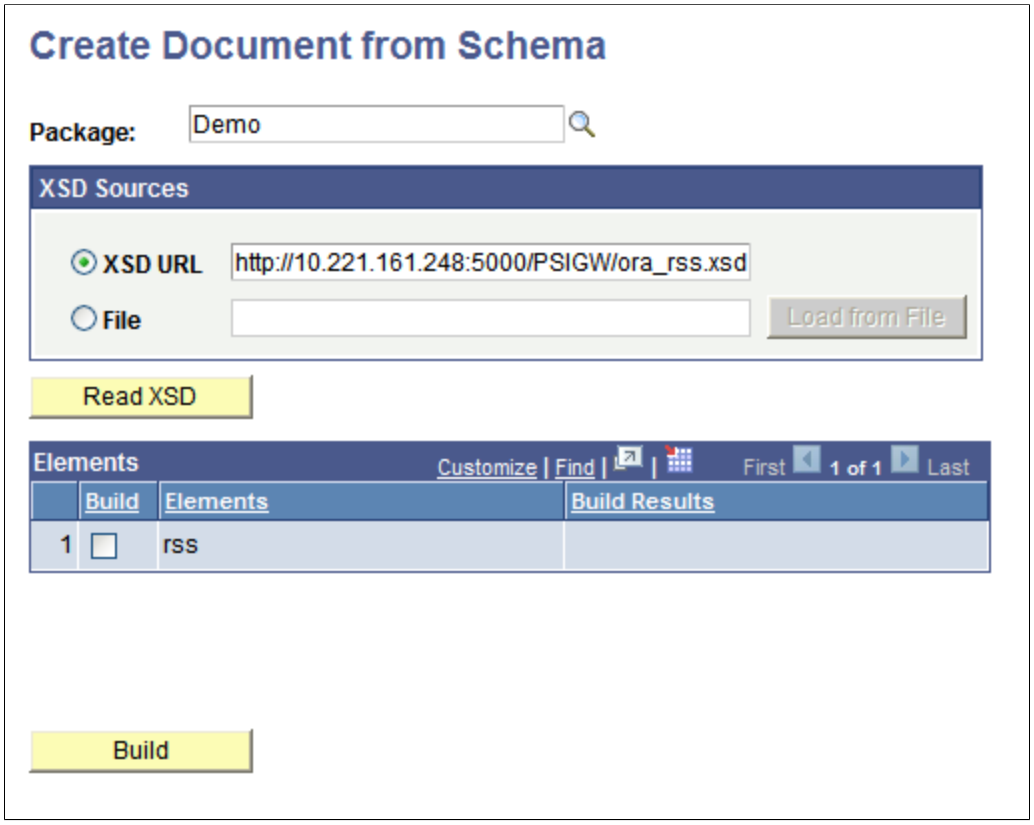
1. Access the Create Document from Schema page (**PeopleTools** > **Documents** > **Document Utilities** > **Create Document from Schema**).
2. In the **Package** field, specify a package for the document you are creating using one of the following methods:
   - Enter a new package name.
   - Enter an existing package name or use the **Lookup** button to select one.
3. Select the source of the schema using one of the following methods:
   - In the **XSD URL** field, enter a schema URL.
   - In the **File** field, enter the schema file name or click the **Load from File** button to browse to and select the file.
4. Click the **Read XSD** button.

An Element section appears at the bottom of the page. The next step is to build the document, as discussed in the next section.

Building Documents from Schema

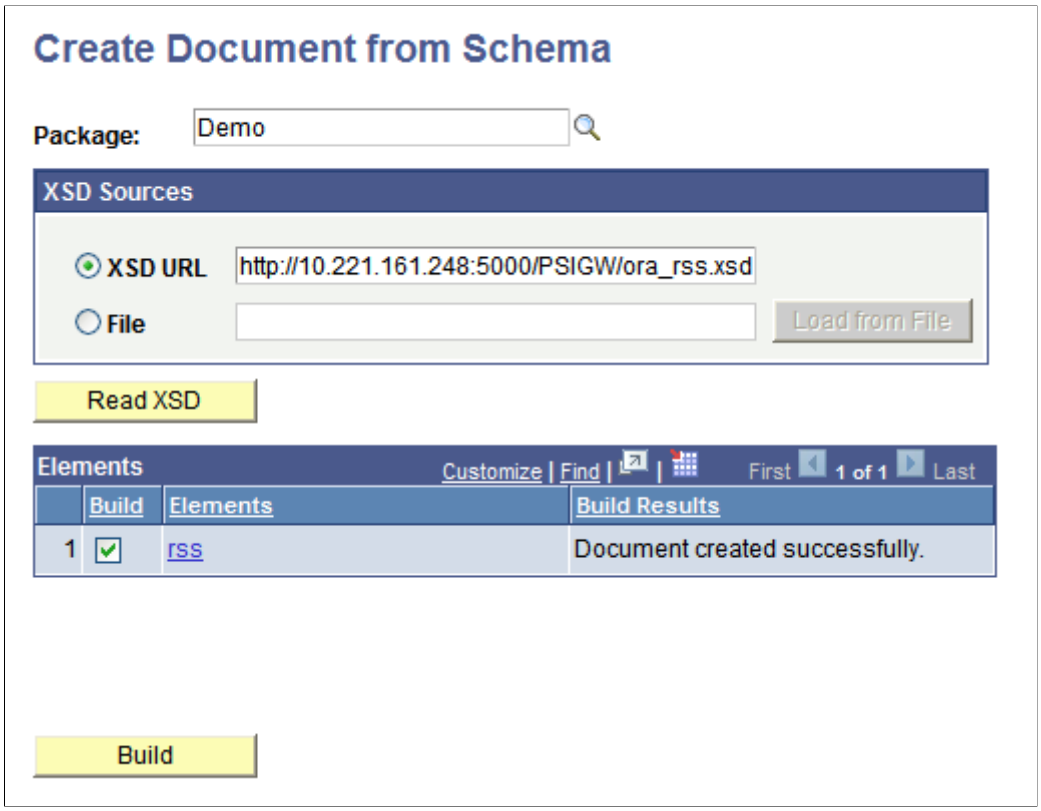
After you specify a package and have read a schema into the system, you select the elements to include in the document and build the document.

The following example shows the Create Document from Schema page after you have read the schema.



After you have read the schema, an Elements section appears at the bottom of the page. Here, you select the elements to build out in the document.

The following example shows the Create Document from Schema page after you have selected the elements to include in the document.



To build a document from schema:

1. In the Elements section, select the **Build** check box for each element to include in the document.
2. Click the **Build** button.

The Build Results column in the Elements section displays the status of the action. The next section discusses how to view and manage the new document.

Viewing Documents Created from Schema in the Document Builder

After you create a document using the Create Document from Schema utility, you can access the document in the Document Builder from the utility.

After you successfully build a document, click the element name in the Elements section to open the document in the Document Builder.

The following example shows the Elements section of the Create Document from Schema page with the **rss**element selected:

When you click the link, the document opens in the Document Builder, and you can view and manage the document like any other document in the system.

The following example shows a partial view of the *rss* document in the Document Builder – Document page.

Chapter 10

Creating Documents from PeopleSoft Records

Understanding Creating Documents from PeopleSoft Records

You can create a document from a PeopleSoft record. When you do so, the record name becomes the root element of the document and the record fields become document elements.

Use the Create Document from Record page to create document from a PeopleSoft record.

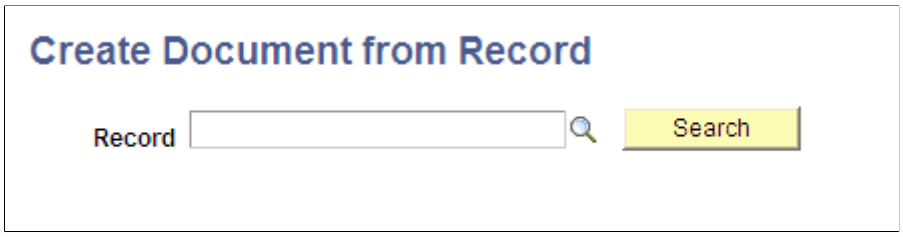
When you create a document from a PeopleSoft record, the record name becomes the root element of the document and the record fields become primitive elements.

Using the Create Document from Record Page

Use the Create Document from Record page (IB_DOCUMENT_CRREC) to select the PeopleSoft record to use to create a document..

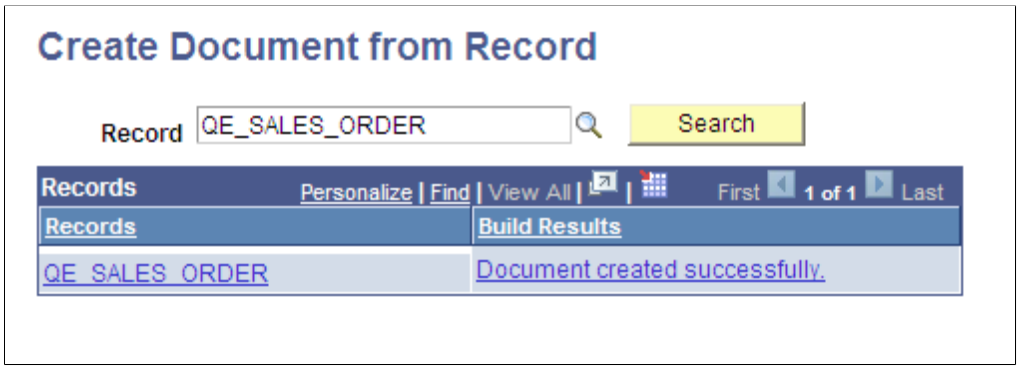
To access the page select **PeopleTools** > **Documents** > **Document Utilities** > **Create Document from Record.**

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



After you select the record to use to create the document and select the record fields to include in the document, the Create Document from Record page appears again and contains two additional links. One of the links enables you to view the document in the Document Builder. The other link provides access to the Select Fields to Insert into New Document page should you want to modify the fields to include as elements in the document.

This example illustrates the Create Document from Record page after a document has been created based on the *QE_SALES_ORDER* record.



The following fields and controls appear on the Create Document from Record page:

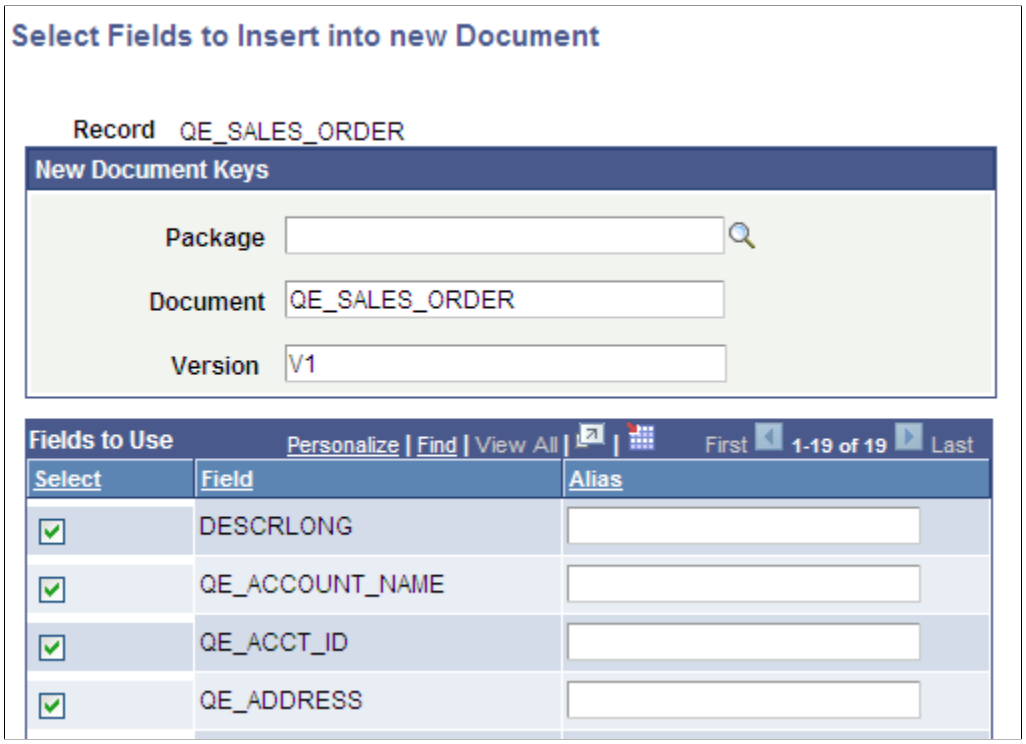
Field or Control	***Description***
Record	Enter the name or use the Lookup button to search for the record to use to create the document.
Search	Click the button to search the database for the record.
Records (grid)	This grid show the search results and build results.
Records	When searching for a record to use to build a document, this field shows the search results. Click the link to access the Select Fields to Insert into New Document page to select the fields to include in the document. After you've created a document based, this field shows the record used to create the document. Click the link to access the Select Fields to Insert into New Document page if you want to modify any of the fields to include in the document.
Build Results	This field appears only after you're created a document based on a record. This field shows the results of building the document based on the record. Click the link to access the document in the Document Builder.

Using the Select Fields to Insert into New Document Page

Use the Select Fields to Insert into New Document page (IB_LOGICALFLD_SEC) to select the record fields to insert into the document and to optionally assign alias names to the fields.

To access the page select **PeopleTools** > **Documents** > **Document Utilities** > **Create Document from Record.** After you search for a record, click the record name in the Records grid.

This example illustrates the fields and controls on the Select Fields to Insert into New Document page. You can find definitions for the fields and controls later on this page.



The example shows a partial view of the Select Fields to Insert into New Document page for the record *QE_SALES_ORDER*.

The following fields and controls appear on the page:

Field or Control	***Description***
Package	Enter the package name for the document or click the Lookup button to search for one.
Document	By default the system uses the record name for the document name. To change the default value, enter a name for the document.
Version	By default the system versions the document as *V1*. To change the default value, enter a version number for the document.
Fields to Use (grid)	Use the fields to use grid to select the fields to include as primitive elements in the document and to optionally assign alias names to the fields.

Field or Control	*Description*
Select	Select the box next to each field to include as a primitive element in the document. By default all fields are selected and included as primitive elements in the document. To select a field, check the box. To deselect a field, check the box again.
Field	Name of the record field.
Alias	(Optional.) Enter an alias for the field.
OK	Click the button to save the changes and return to the Create Document from Record page.
Cancel	Click the button to return to the Create Document from Record page without saving the changes.

Creating Documents from Records

To create a document from a PeopleSoft record:

1. Access the Create Document from Record page (**PeopleTools** > **Documents** > **Document Utilities** > **Create Document from Record.**
2. In the Record field, enter the name of the record to use for the document or click the **Lookup** button to search for one.
3. Click the **Search** button.

   The search results appear in the Record grid.
4. Click the name of the record to use for the document.

   The Select Fields to Insert into New Document page appears.
5. In the **Package** field, enter a the package name for the document or click the **Lookup** button to search for one.
6. In the **Document** field, enter the name of the document.

   The default value is the record name. You can use the default value or enter a new value.
7. In the **Version** field, enter the document version.

   The default value is *V1*. You can use the default value or enter a new value.

8. In the Fields to Use grid, select the Select box for each fields to include as a primitive element in the document.

   By default, all the fields in the record are selected. To deselect a field, select the box. To select a field, select the box again.

9. (Optional.) In the Alias field enter an alias for a field.

10. Click the **OK** button.

Viewing Documents Created from Records

After you created a document from a record you can view the document in the Document Builder.

To view the document in the Document Builder, in the Create Document from Record page, in the **Build Results** field, click the **Document Created Successful** link. The document appears in the Document Builder.

This example illustrates the document created from the *QE_SALES_ORDER* record appearing in the Document Builder.

In addition to navigating to view the document in the Document Builder from the Create Document from Record page, you can also use the standard PIA navigation path to the Document Builder. The standard PIA path is **PeopleTools** > **Documents** > **Document Builder.**

Chapter 11

Copying and Exporting Documents

Understanding Copying and Exporting Documents

The Document Builder enables you to make a copy of an existing document, as well as export document schema to a file.

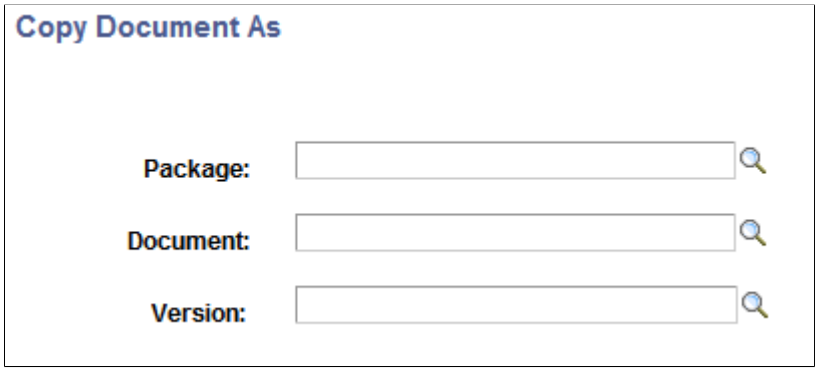
However, if a document is read-only, you cannot copy or export it. The conditions under which a document is read-only are described elsewhere in the product documentation.

See Managing Write-Access to Documents.

Copying Documents

Use the Documents Builder copy feature to copy and save a document using a different name. When you click the **Copy** button from the Documents Builder - Document tab, the following Copy Document As page (IB_LOGICALCOPY) appears:

Use the Copy Document As page to make a copy of a document.



When you copy a document, one of the following parameters you use for the new document must be different than the original: Package, Document, or Version.

To copy a document:

1. Access the Document Builder (**PeopleTools** > **Documents** > **Document Builder**).
2. Locate and open the document definition to copy.

   The document appears in the Document Builder.
3. At the bottom of the Document Builder page, click the **Copy** button.

   The Copy Document As page appears.

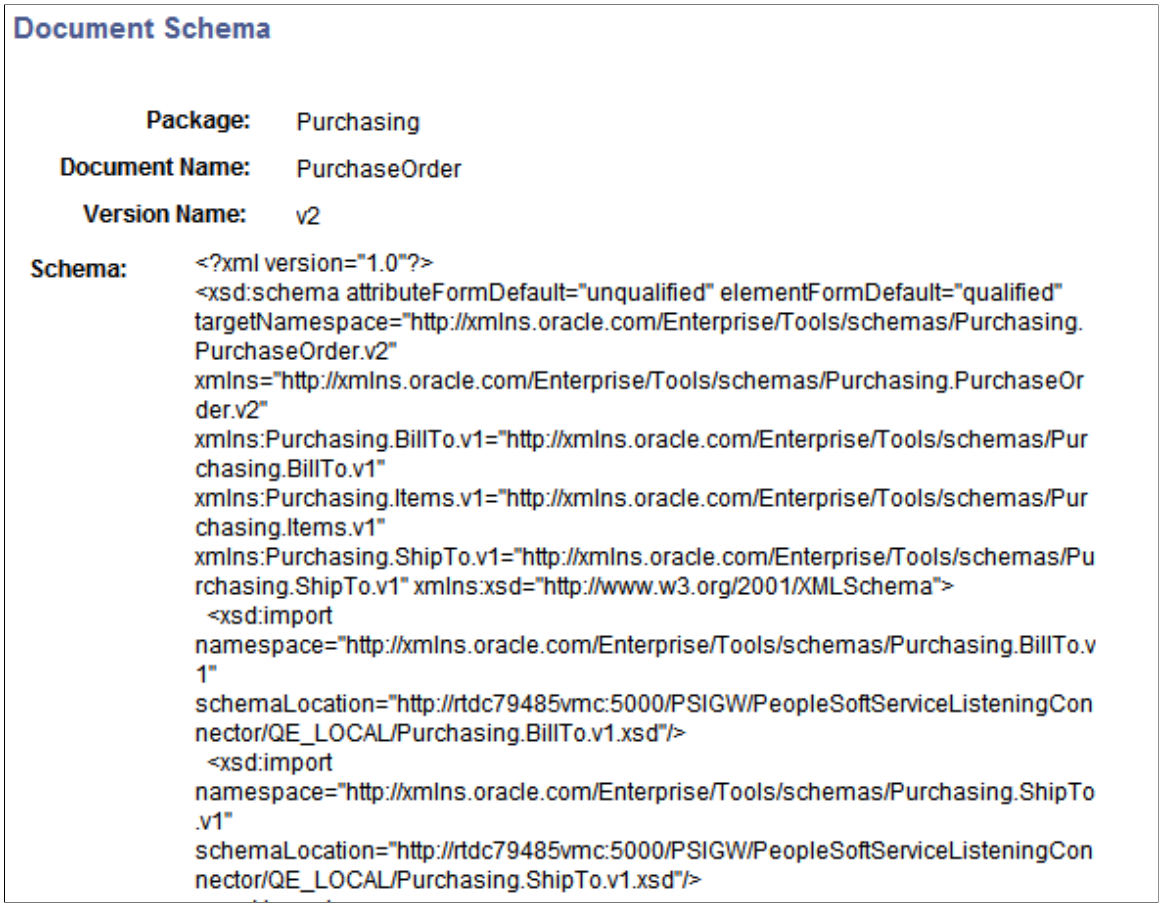
4. In the **Package** field, enter the package name or click the **Lookup** button to search for one.
5. In the **Document** field, enter the document name or click the **Lookup** button to search for one.
6. In the **Version** field, enter the version number or click the **Lookup** button to search for one.
7. Click the **OK** button.

   The new copied version of the document appears in the Document Builder.

Exporting Documents

The Document Builder provides a Document Schema page (IB_LOGICALSCMA_SEC) that enables you to export an XML document schema to a file. To access the page, click the **Export** button at the bottom of the Document Builder - Documents page.

The following example shows the Document Schema page.



Document Schema

Package: Purchasing

Document Name: PurchaseOrder

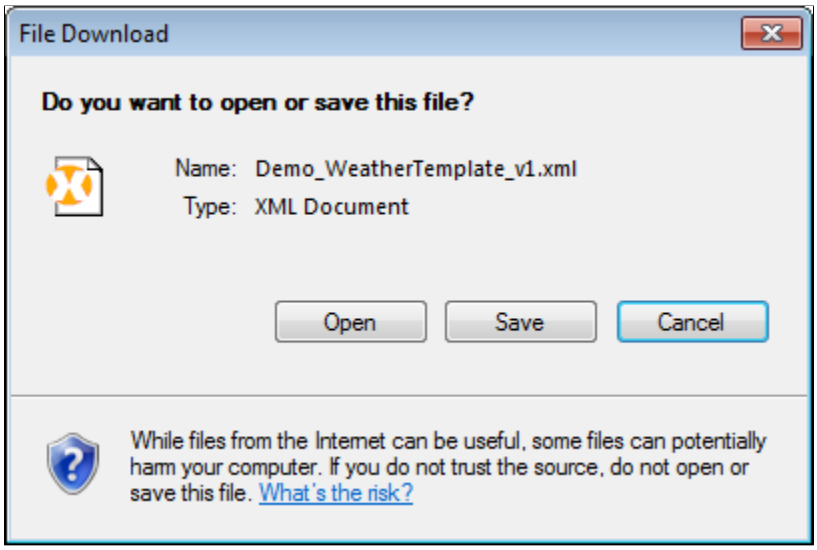
Version Name: v2

Schema:

```
<?xml version="1.0"?>
<xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/Enterprise/Tools/schemas/Purchasing.
PurchaseOrder.v2"
xmlns="http://xmlns.oracle.com/Enterprise/Tools/schemas/Purchasing.PurchaseOr
der.v2"
xmlns:Purchasing.BillTo.v1="http://xmlns.oracle.com/Enterprise/Tools/schemas/Pur
chasing.BillTo.v1"
xmlns:Purchasing.Items.v1="http://xmlns.oracle.com/Enterprise/Tools/schemas/Pur
chasing.Items.v1"
xmlns:Purchasing.ShipTo.v1="http://xmlns.oracle.com/Enterprise/Tools/schemas/Pu
rchasing.ShipTo.v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:import
namespace="http://xmlns.oracle.com/Enterprise/Tools/schemas/Purchasing.BillTo.v
1"
schemaLocation="http://rtdc79485vmc:5000/PSIGW/PeopleSoftServiceListeningCon
nector/QE_LOCAL/Purchasing.BillTo.v1.xsd"/>
 <xsd:import
namespace="http://xmlns.oracle.com/Enterprise/Tools/schemas/Purchasing.ShipTo
.v1"
schemaLocation="http://rtdc79485vmc:5000/PSIGW/PeopleSoftServiceListeningCon
nector/QE_LOCAL/Purchasing.ShipTo.v1.xsd"/>
```

A **Save As** button appears at the bottom of the Document Schema page. When you click the button, a File Download dialog box appears.

This example shows the File Download dialog box used to export a document schema from the Document Builder.



You can save the schema to a file or open it in an XML editor.

The XML schema of the document saved in the database is the schema that the system exports; therefore, you must save any updates to the document so that they are reflected in the exported XML schema.

The export feature exports the entire document definition.

To export a document:

1. Access the Document Builder (**PeopleTools** > **Documents** > **Document Builder**).
2. Locate and open the document to export.

   The document appears in the Document Builder.
3. At the bottom of the Document Builder page, click the **Export** button.

   The Document Schema page appears.
4. At the bottom of the Document Schema page, click the **Save As** button.

   A File Download dialog box appears.
5. Perform one of the following actions:
   - Click the **Save** button to save the schema to file.
   - Click the **Open** button to open the schema in an XML editor of your choice.
6. Close the File Download dialog box.
7. On the Document Schema page, click the **Return** button to go back to the Document Builder - Document page.

Copying Documents Between Databases

This section discusses how to:

- Copy documents between PeopleTools 8.52 and later databases.
- Copy documents between PeopleTools 8.51 databases.

Copying Documents Between PeopleTools 8.52 and Later Databases

You can use PeopleSoft Application Designer's Project Copy functionality to copy document metadata between PeopleTools 8.52 and later databases.

Related Links

“Copying Projects” (Lifecycle Management Guide)

Copying Documents Between PeopleTools 8.51 Databases

If you copy a document schema from a later release to a PeopleTools 8.51 database, schema validation may fail. Delete the schema before you copy the document to the PeopleTools 8.51 database; the system automatically rebuilds document schema when you save the document or deploy it.

There are no restrictions for copying documents and associated document schema from a PeopleTools 8.51 database to a later release

Related Links

“Copying Projects” (Lifecycle Management Guide)

Chapter 12

Renaming and Deleting Documents

Understanding Renaming and Deleting Documents

If a document is read-only, you cannot rename or delete it. The conditions under which a document is read-only are described elsewhere in the product documentation.

See Managing Write-Access to Documents.

Renaming Documents

This section provides an overview of renaming documents and discusses how to:

- Rename documents in the Document Builder.
- Rename documents in the Document Administration component.

Understanding Renaming Documents

You can rename a document in two locations in the PeopleSoft system:

- Document Builder.
- Document Administration component.

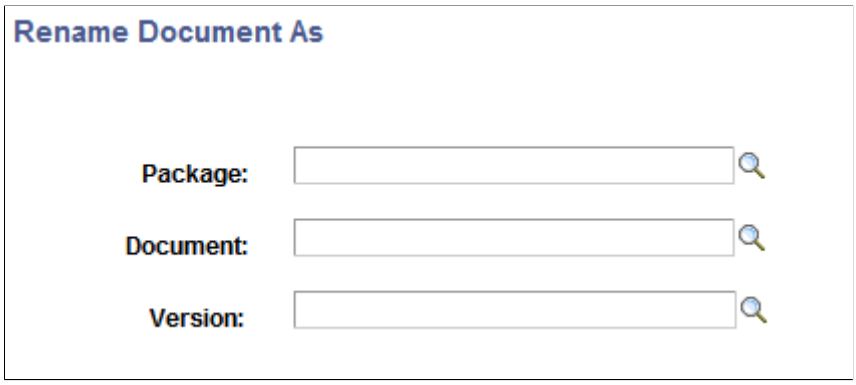
This section describes how to rename documents in both locations.

When you rename a document, at least one of the following parameters must be different than the original: Package, Document, or Version.

Renaming Documents in the Document Builder

You can rename a document in the Document Builder, using the Rename Document As page (IB_LOGICALCOPY). To access the page, select **PeopleTools** > **Documents** > **Document Builder** and click the **Rename** button.

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



To rename a document in the Document Builder:

1. Access the Rename Document As page (**PeopleTools** > **Documents** > **Document Builder** and click the **Rename** button).

   The Rename Document As page appears.

2. Rename the document by selecting values for the following fields. Keep in mind that at least one of the fields must be changed to a value different than the original document name:

   a. In the **Package** field, enter a package name or click the **Lookup** button to search for one.

   b. In the **Document** field, enter a document name or click the **Lookup** button to search for one.

   c. In the **Version** field, enter a version number or click the **Lookup** button to search for one.

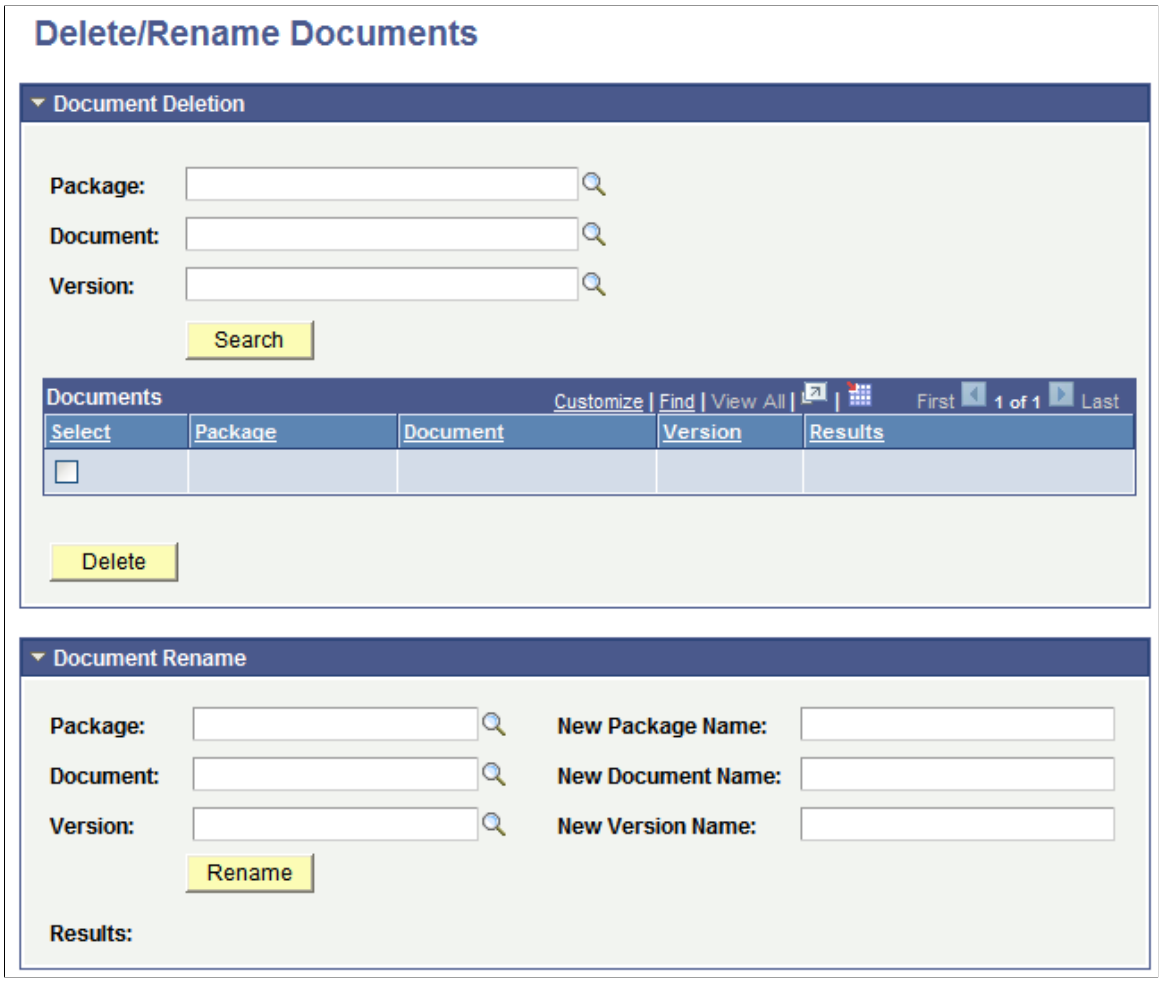
3. Click the **OK** button.

   The renamed version of the document appears in the Document Builder - Document page.

Renaming Documents in the Document Administration Component

You can also rename a document in the PeopleSoft system using the Delete/Rename Document page (IB_DOCUMENT_ADMIN) located in the Document Administration component. To access the page, select **PeopleTools** > **Documents** > **Document Administration** > **Delete or Rename Document**.

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



Use the Document Rename section at the bottom of the page to rename a document.

To rename a document in the Document Administration component:

1. Access the Delete or Rename Document page (**PeopleTools** > **Documents** > **Document Administration** > **Delete or /Rename Document**).

   The Delete/Rename Document page appears.

2. At the bottom of the page, click the arrow icon next to Document Rename to expand the section if it is not already expanded.

3. On the left side of the section, enter the details of the existing document to rename:

   a. In the **Package** field, enter the package name or click the **Lookup** button to search for one.

   b. In the **Document** field, enter the document name or click the **Lookup** button to search for one.

   c. In the **Version** field, enter the version number or click the **Lookup** button to search for one.

4. On the right side of the section, enter the new naming details of the document. Keep in mind that at least one of the values you enter must be different than the original name:

    a. In the **New Package Name** field, enter a package name.

    b. In the **New Document Name** field, enter a document name.

    c. In the **New Version Name** field, enter a version number.

5. Click the **Rename** button.

    The **Results** field at the bottom of the page indicates if the action was successful.

Deleting Documents

This section provides an overview of deleting documents and discusses how to:

- Delete documents in the Document Builder.
- Delete documents in the Document Administration Component.

Understanding Deleting Documents

You can delete documents from two locations in the PeopleSoft system: in the Document Builder and on the Delete/Rename Documents page of the Document Administration component.

You can use either location to delete a single document from the system. To delete multiple documents at a time, use the Delete/Rename Documents page.

Note that when you delete a document, you delete the document and all related objects.

Deleting Documents in the Document Builder

To delete a document in the Document Builder:

1. Access the Document Builder - Document page (**PeopleTools** > **Documents** > **Document Builder).**
2. Click the **Delete** button.
3. Click the **OK** button.

Deleting Documents in the Document Administration Component

You can also delete a document in the PeopleSoft system using the Delete/Rename Document page (IB_DOCUMENT_ADMIN) located in the Document Administration component. Using the Delete/Rename page, you can delete a single document or multiple documents.

To delete a document in the Document Administration component:

1. Access the Delete/Rename Document page (**PeopleTools** > **Documents** > **Document Administration** > **Delete/Rename Document**).

   The Delete/Rename Document page appears.

2. Click the arrow icon next to the Document Deletion label to expand the section if it is not already expanded.

3. Search for the documents to delete.

   See Searching for Document Definitions.

4. In the Documents section, select the Select check box next to each document that you want to delete.

5. Click the **Delete** button.

Chapter 13

Securing Documents

Understanding Securing Documents

PeopleSoft provides several options for securing documents at the definition level.

Note: Along with exploring the options described in this topic, a security analyst should assess your overall security requirements.

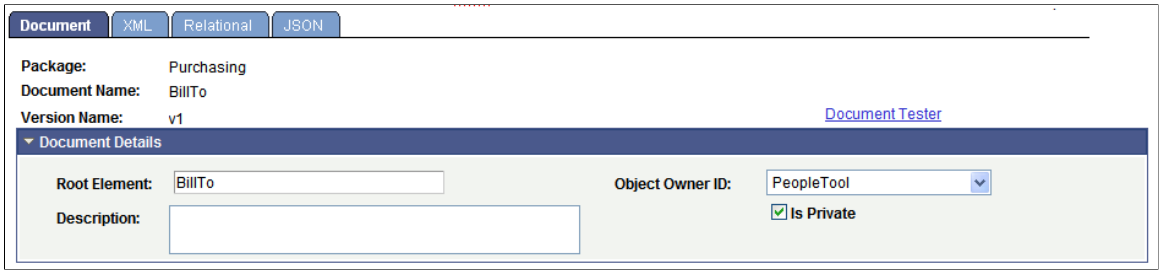
You can specify that documents be used only in the document package to which they are defined. In doing so, the document is not an option for documents belonging to other packages to reference. In addition, you can restrict write-access to documents, making them read-only.

Specifying Private Documents

When you specify a document as private, the document can only be used in its own document package and can only be referenced by other documents in the same package.

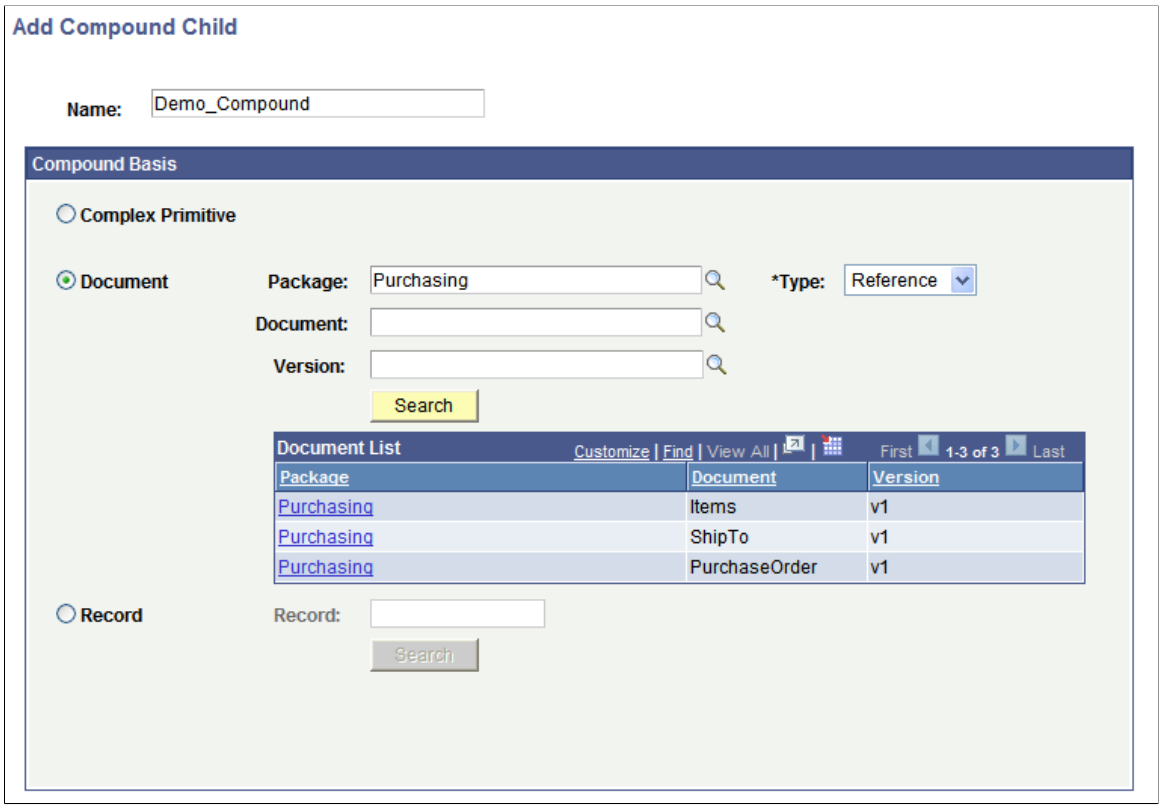
You specify a document as private by selecting the **Is Private** check box in the Document Details section of the document definition.

The following example shows the Document Details section of the *BillTo* document:



When you mark a document private, the document does not appear as a reference option when attempting to specify it in a document that belongs to any other document package than its own.

The following example shows the Add Compound Child page for the document *Demo_Document* that belongs to the *Demo* package.



The previous example shows that the when searching for a document in the Purchasing package to reference, the BillTo document does not appear as an option, since it is designated a private document.

To specify a private document:

1. Select **PeopleTools** > **Documents** > **Document Builder.**
2. Click the arrow icon next to the Document Details label to expand the section.
3. Select the **Is Private** check box.
4. Click **Save.**

Managing Write-Access to Documents

This section discusses how to:

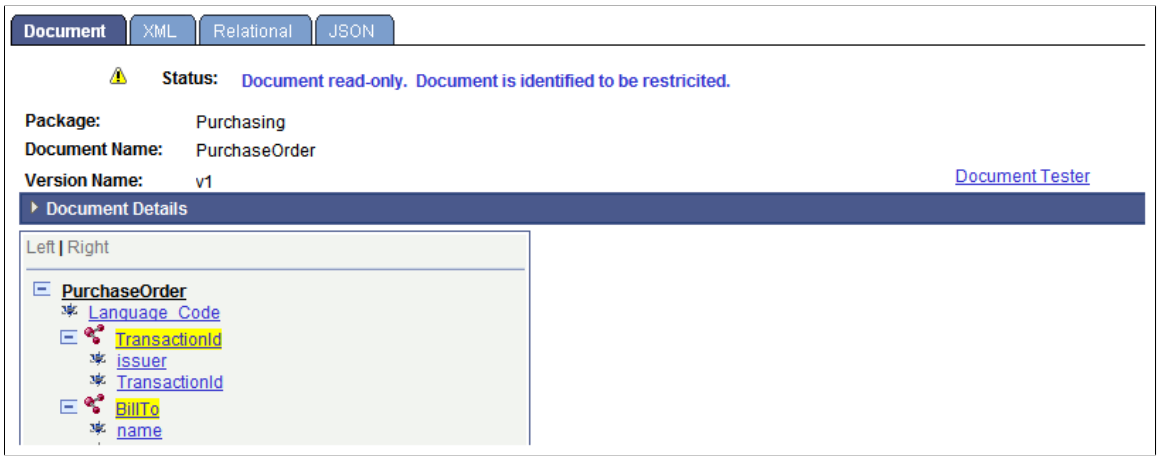
- Restrict write-access to documents.
- Clear restricted write-access to documents.

Restricting Write-Access to Documents

When you restrict write-access to a document, it becomes read-only.

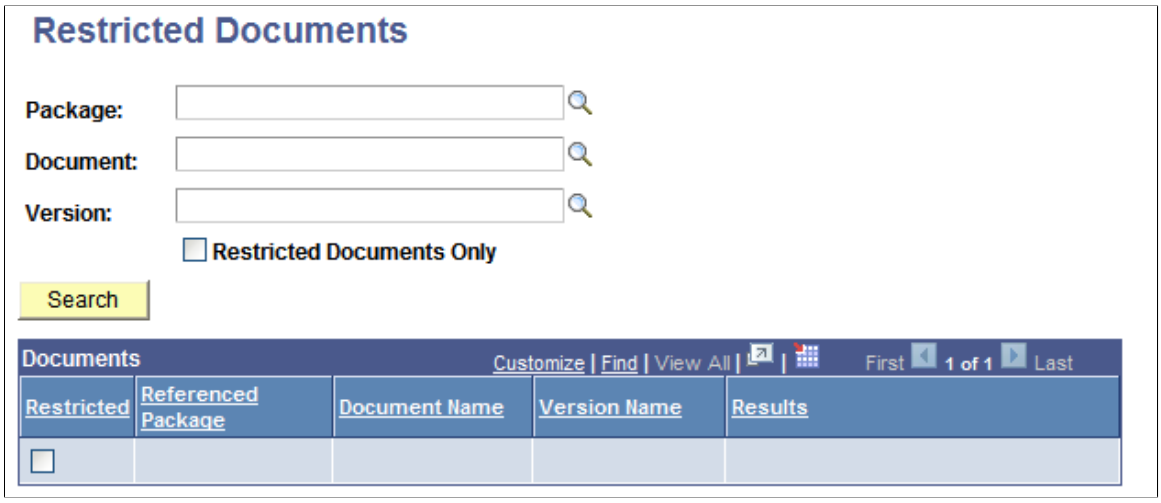
When someone accesses a document that is read-only, a status message appears at the top of the document definition that indicates it is a restricted document and is read-only.

The following example shows a document with restricted write-access. The status message at the top of the page indicates that the *PurchaseOrder* document is a restricted document.



Use the Restricted Documents page (IB_DOCUMENT_RSTRT) to restrict write-access to documents. To access the page, select **PeopleTools** > **Documents** > **Document Administration** > **Restricted Documents.**

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



To restrict write-access to documents:

1. Access the Restricted Documents page (PeopleTools, Documents, Document Administration, Restrict Documents).
2. Search for the document or documents for which to restrict write-access.

   See Searching for Document Definitions.

   The search results appear in the **Documents** grid at the bottom of the page.

3. Select the **Restricted** check box next to each document for which to restrict write-access.
4. At the bottom of the page, click the **Update** button.

   In the Results column next to each selected document, the message *Document restricted* appears.

Clearing Restricted Write-Access to Documents

To clear restricted write-access to documents, use the Restricted Documents page (IB_DOCUMENT_RSTRT) shown in the previous section.

To clear restricted write-access to documents:

1. Access the Restricted Documents page (**PeopleTools** > **Documents** > **Document Administration** > **Restricted Documents**).
2. Search for the document or documents for which to clear restricted access.

   a. To search for a specific document, enter one or more values in the following fields: **Package**, **Document**, **Version**.

   b. To display all restricted documents in the system, select the **Restricted Documents Only** check box.

3. Click the **Search** button.

   The search results appear in the **Documents** grid.

   Observe that the **Restricted** check box is selected next to all documents with restricted access.
4. Clear the **Restricted** check box next to each document for which to clear restricted write-access.
5. At the bottom of the page, click the **Update** button.

   In the **Results** column, the message *Document unrestricted* appears next to each document for which you cleared restricted access.

Chapter 14

Testing Document Schema

Understanding Testing Document Schema

PeopleSoft provides a Document Schema Tester utility that enables you to test and validate XML document schema generated by the Document Builder.

The Document Schema Tester utility enables you to validate XML documents against document schemas during development to determine if documents adhere to defined document schemas.

Prerequisites for Testing Document Schema

To use the Document Schema Tester utility and test document schema, the following items must exist:

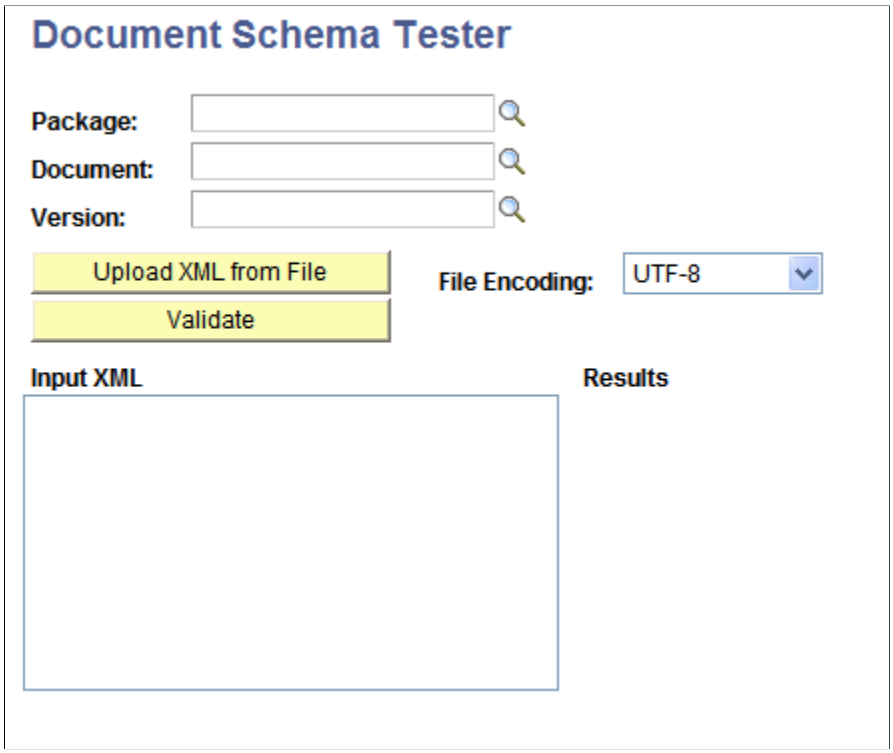
- A document schema against which to test a document.
- An XML document to test against a schema.

Access the Document Schema Tester Utility

The Document Schema Tester utility is located in the Document Schema Tester component (IB_DOCSCHEMATESTER).

To access the Document Schema Tester page (IB_DOCSCHEMATESTER), select **PeopleTools** > **Documents** > **Document Utilities** > **Document Schema Tester.**

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



Testing XML Document Schema

To test XML document schema:

1. Access the Document Schema Tester page (**PeopleTools** > **Document Utilities** > **Document Schema Tester**).
2. Select the document against which to test the schema:
    a. In the **Package** field, enter the name of the package to which the document belongs or click the **Lookup** button to search for the name.
    b. In the **Document** field, enter the name of the document or click the **Lookup** button to search for the name.
    c. In the **Version** field, enter the document version or click the **Lookup** button to search for the version.
3. From the **File Encoding** drop-down list box, select the file encoding of the XML schema you are loading. The options are:
    - *Non-Unicode.*
    - *UTF-8.* (Default.)

- *UTF-16.*

4. Load the XML document schema to test against the document. Use one of the following options:
   - To load an XML document schema from file, click the **Load XML from File** button. The File Attachment dialog box appears. Click the **Browse** button to locate the XML document schema to upload and click the **Upload** button.
   - To load an XML document schema manually, enter the XML schema in the **XML Input** field.
5. Click the **Validate** button.

The results of the test appear in the Results area of the page.

Chapter 15

Testing Documents

Understanding Testing Documents

PeopleSoft provides a Document Tester utility that enables you to test the physical format of documents outside of runtime.

Overview

The Document Tester utility provides you the option of entering sample data values for primitive elements, as well as appending or deleting rows in collection elements. You can clear test values and actions at anytime, and assign new test values and actions and regenerate the test document.

Use the Document Tester to generate a physical document. The system then verifies if the format is valid and if it can be parsed. In addition, the system displays the generated document for you to view.

The Document Tester also provides a link to the Document Builder, should you need to make changes to a document as a result of testing it.

Note the following points about the Document Tester utility:

- You can generate test documents with or without entering test data values or performing append and delete actions.
- You cannot use the Document Tester to test CDATA type primitive elements.

Physical Documents Generated

The Document Tester provides the option to generate documents in the following physical formats:

- XML.
- JSON.
- PeopleCode.

When you generate a document in the PeopleCode format, the system generates the PeopleCode to populate the document. You can cut the PeopleCode and paste it anywhere needed to populate the document. You need only supply the actual data for each element. Note that for collections, the code only shows one row of data (single create and append item).

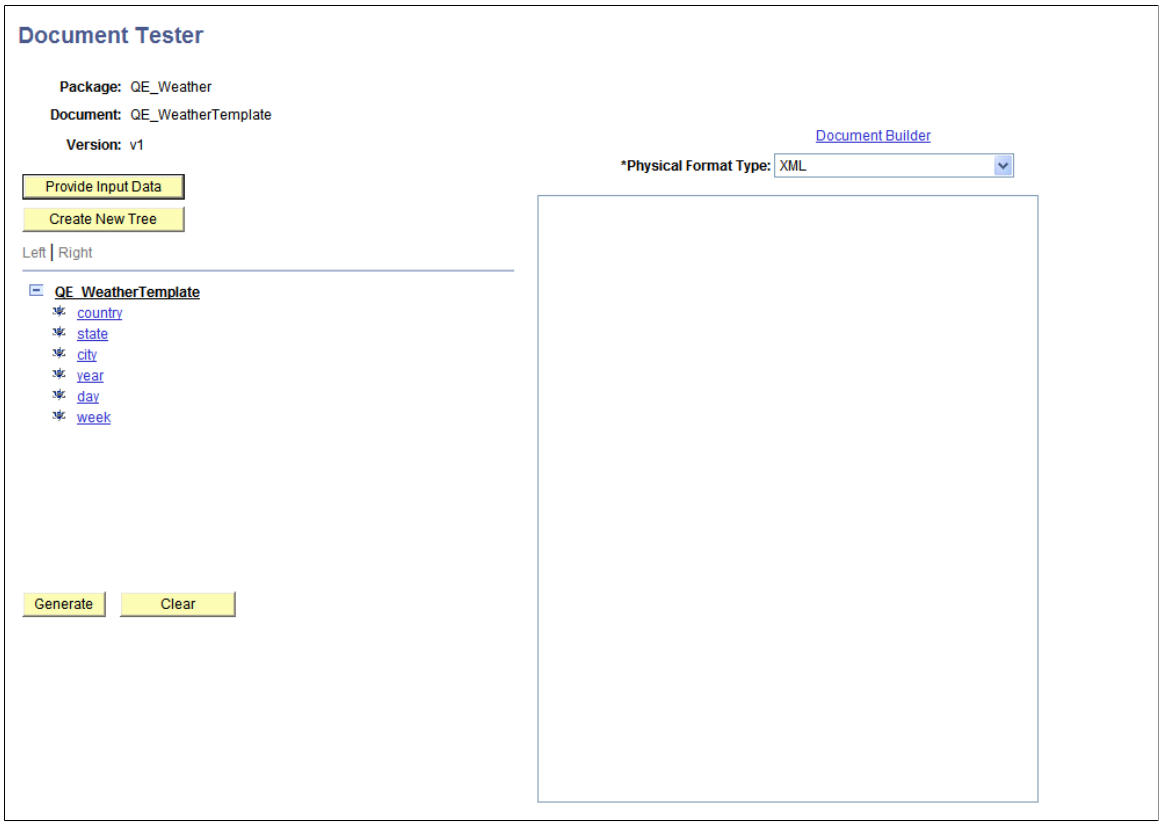
Accessing the Document Tester Utility

The Document Tester utility page (IB_LOGICALTESTER) is located in the IB_LOGICALTESTER component.

You can access the Document Tester utility page two ways:

- From the **Document Tester** link in the Document Builder (**PeopleTools** > **Documents** > **Document Builder**).
- Select **PeopleTools** > **Documents** > **Document Utilities** > **Document Tester.**

The following example shows the Document Tester page.



Entering Element Test Values

This section discusses how to:

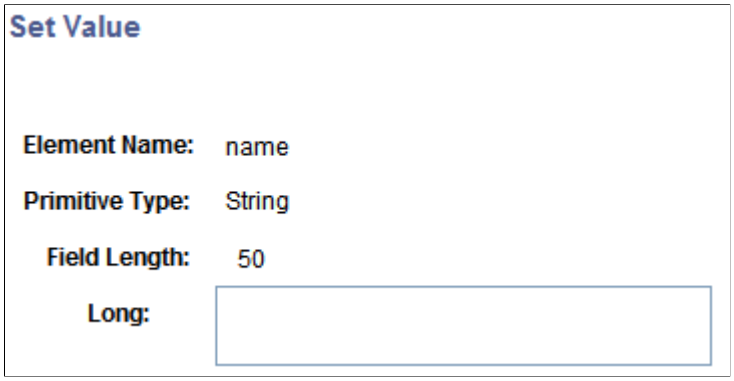
- Enter test values for primitive elements.
- Append and delete collection element items.

Entering Test Values for Primitive Elements

The Document Tester utility enables you to enter test values for primitive elements.

When you select a primitive element in the document tree, a Set Value page (IB_LSTESTER_SEC) appears where you can set a test value for the element.

The following example shows the Set Value page.



The Element Name field displays the element name with which you're working. The Primitive Type field and Field Length field display the data type and length as defined for the element in the Document Builder.

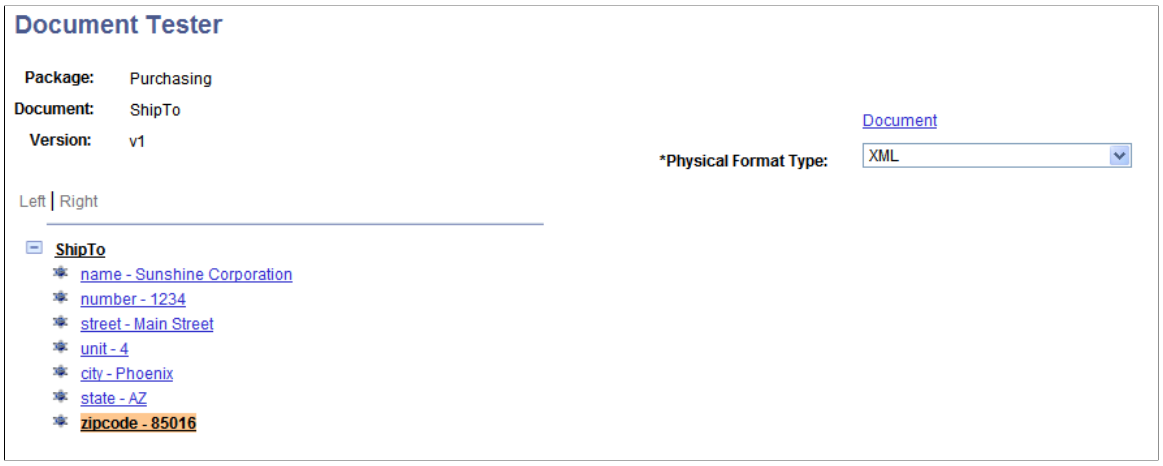
The name of the field where you enter a test value depends on the data type. In the previous example, the data type is a string, and therefore the system prompts you to enter a Long value. The following table lists the possible labels for the field where you enter a test value:

Data Type	***Primitive Type Field Label***	***Test Value Field Label***
Binary	Bin	Long
Boolean	Bool	Page displays a check box.
Character	Character	Char
Date	Date	Date
DateTime	DT	Datetime
Decimal	Dec	Numeric
Integer	Int	Numeric
String	String	Long
Text	Text	Long
Time	Time	Time

The data type and field length that display are those that are defined for the document in the Document Builder.

After you enter test values, each value appears next to the corresponding element in the document tree on the main Document Tester page.

The following example shows the appearance of the document tree for the *ShipTo* document after test values have been entered:



To enter test values for primitive elements:

1. Access the Document Tester (**PeopleTools** > **Documents** > **Document Utilities** > **Document Tester**).
2. Click the name of a primitive element for which to enter a test value.

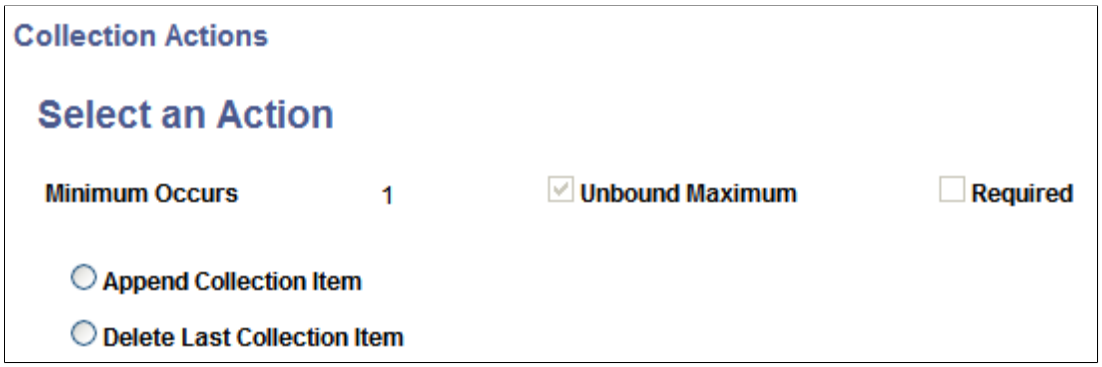
   The Set Value page appears.
3. In the **Long** field, enter a test value.
4. Click the **OK** button.

Appending and Deleting Collection Element Items

You can append a collection item with a copy of the last row in the collection or delete the last row in a collection.

To accomplish either task, use the Collection Actions - Select an Action page (IB_LSTESTER2_SEC). When you click the name of a collection element in the document tree, the Collection Actions - Select an Action page appears.

This example illustrates the fields and controls on the Collection Actions - Select an Action page. You can find definitions for the fields and controls later on this page.



The system displays properties for the element, such as minimum occurs, unbound maximum, and required, as defined for the document in the Document Builder.

Appending Collection Elements

To append a collection element with a copy of the last row in the collection:

1. Access the Document Tester (**PeopleTools** > **Documents** > **Document Utilities** > **Document Tester**).
2. Click the name of a collection element.

   The Collection Actions - Select an Action page appears.
3. Select **Append Collection Item.**
4. Click the **OK** button.

The Document Tester page appears, and the collection element is appended with a copy of the last row in the collection.

Deleting Collection Elements

To delete a collection row:

1. Access the Document Tester (**PeopleTools** > **Documents** > **Document Utilities** > **Document Tester**).
2. Click the name of a collection element.

   The Collection Actions - Select an Action page appears.
3. Select **Delete Last Collection Item.**
4. Click the **OK** button.

The Document Tester page appears, and the last row in the collection is deleted.

Testing Raw Data

This section discusses how to:

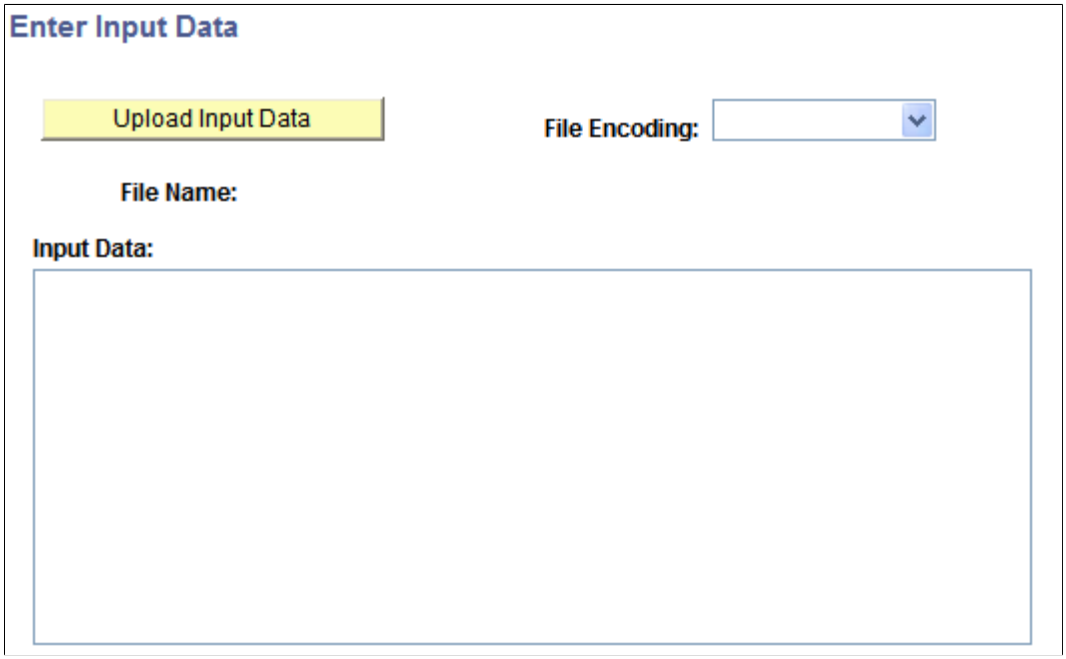
- Upload raw test data from a file.
- Manually enter raw test data.

Uploading Raw Test Data from Files

PeopleSoft enables you to upload raw document data from files and test it in the Document Tester.

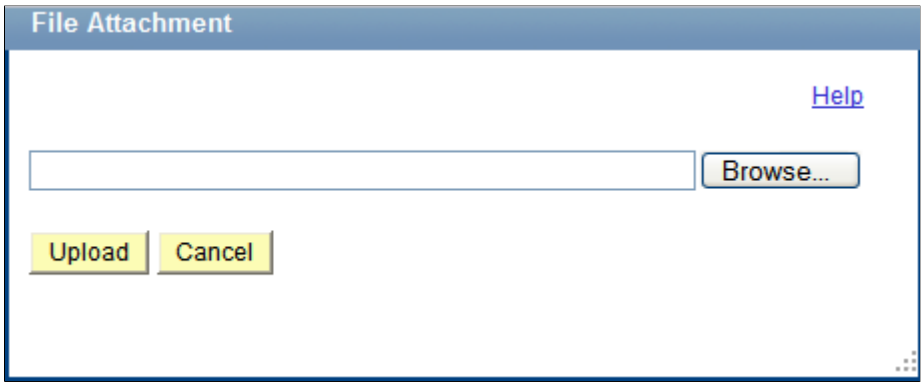
The Document Tester features an Enter Input Data page (IB_LOGTESTER_SEC) that you use to upload raw test data from files. To access the page select **PeopleTools** > **Documents** > **Document Tester** and click the **Enter Input Data** button.

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



After you access Enter Input Data page you click the **Upload Input Data** button to specify the file to upload and test. You use the File Attachment page (IB_LOGTESTER_SEC) to upload the attachment:

This example illustrates the fields and controls on the File Attachment page. Use the page to browse for and upload the raw data file to test.



To upload raw test data from a file:

1. Access the Document Tester (**PeopleTools** > **Documents** > **Document Tester.**)
2. Click the **Provide Input Data** button.

   The Enter Data Input page appears.
3. From the **File Encoding** drop-down list, select the file encoding of the file you are uploading. The options are:

   - *Non-encoded.*
   - *UTF-8.*
   - *UTF-16.*
4. Click the **Upload Input Data** button. The File Attachment page appears.
5. Click the **Browse** button to navigate to and select the file to upload.
6. Click the **Upload** button.

   The Enter Data Input page appears and the content of the raw data file appears in the Input Data section of the page.
7. Click the **OK** button to return to the Document Tester page where you generate and view the test results.

Generating and viewing test documents is described in the next section.

See Generating and Viewing Test Documents.

Manually Entering Raw Test Data

To manually enter data to test you use the Enter Data Input page described in the previous section.

To manually enter data to test:

1. Access the Document Tester (**PeopleTools** > **Documents** > **Document Tester**.
2. Click the **Provide Input Data** button.

   The **Enter Data Input** page appears.
3. In the **Input Data** box enter the data to test.
4. Click the **OK** button to return to the Document Tester page where you generate and view the test results.

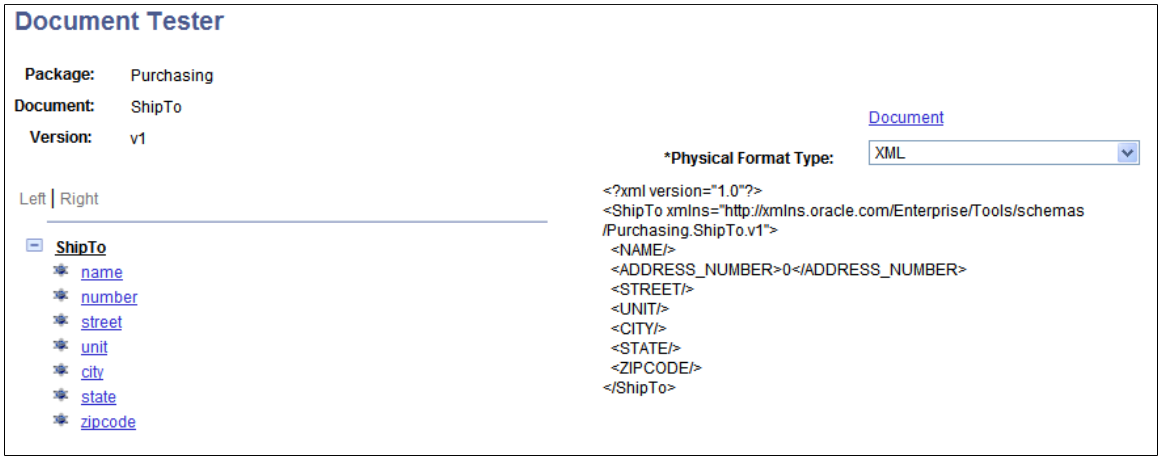
Generating and viewing test documents is described in the next section.

See Generating and Viewing Test Documents.

Generating and Viewing Test Documents

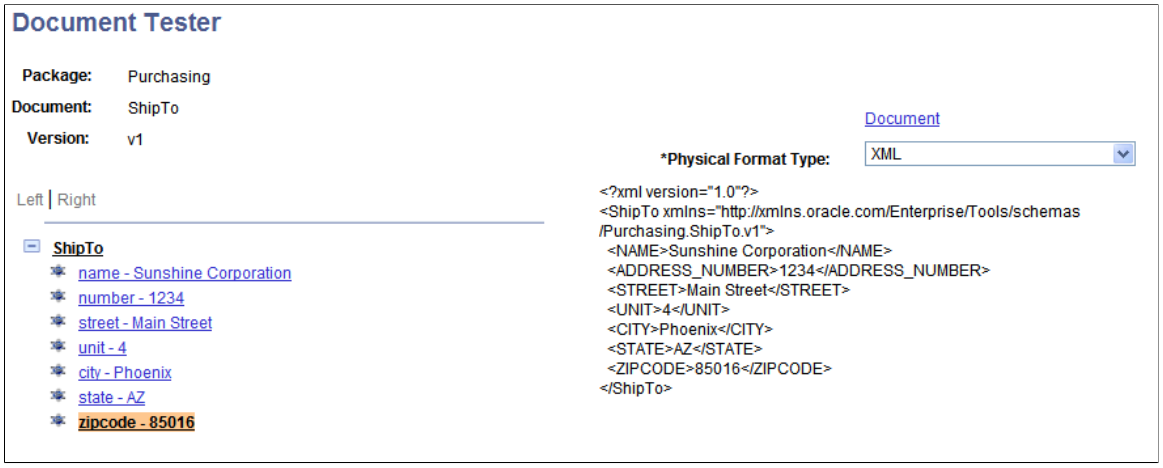
When you generate a test document in the Document Tester, the generated document appears on the right side of the Document Tester page.

The following example shows an XML document generated for the *ShipTo* document:



When you view the output you'll note that no test values were specified for the primitive elements.

The following example shows test output generated for the *ShipTo* document using specified test values:



To generate and view test documents:

1. Access the Document Tester page (**PeopleTools** > **Documents** > **Document Utilities** > **Document Tester**).
2. Enter any optional test values and actions as described elsewhere in this topic.
3. From the **Physical Format Type** drop-down list box, select the document format to generate:
   - **XML.** Select this option to generate the document in XML format.
   - **JSON.** Select this option to generate the document in JSON format.
   - **PeopleCode.** Select this option to generate PeopleCode to populate the document.
4. Click the **Generate** button at the bottom of the page.

The test document in the physical format selected appears on the right side of the page.

Verifying XML and JSON Parsing

Use the Document Tester utility to verify the proper XML or JSON parsing of data and subsequent generation of XML or JSON.

This capability is useful for inbound data that will subsequently be loaded into a document, as you can validate that the data is of a valid format and structure expected based on the format type.

Generating and Parsing XML Structures

The PeopleCode to generate an XML structure is as follows:

```
&string = &DOC.GenXMLString();
// returns an XML string based on the Document.
```

The PeopleCode to parse an XML structure is as follows:

```
&bool = &DOC.ParseXMLString(string &xmldata, <optional> bool &b);
```



```
// The second parameter is optional and will validate the XML data based on
// the XML schema if set to TRUE.
```

Generating and Parsing JSON Structures

The PeopleCode to generate a JSON structure is as follows:

```
&string = &DOC.GenJsonString();
// returns a JSON string based on the Document.
```

The PeopleCode to parse a JSON structure is as follows:

```
&bool = &DOC.ParseJsonString(string &jsondata, <optional>  bool &b);
// The second parameter is optional and will validate the JSON data based on
// the XML schema if set to TRUE.
```

Clearing Document Tester Utility Data

This section discuses how to:

- Clear Document Tester output.
- Clear test values and actions.

Clearing Document Tester Output

To clear generated test output, for example, generated XML, click the **Clear** button at the bottom of the Document Tester utility page.

Clearing Test Values and Actions

This section discusses how to clear primitive element test values and collection element actions.

Clearing Primitive Element Test Values

To clear primitive element test values:

- To clear one or several values, click an element in the document tree to open the Set Value page. Manually clear the value.
- To clear all test values, exit the Document Tester and reopen it.

Clearing Collection Element Actions

To clear collection element actions:

- To clear an appended collection element, click the element in the document tree to open the Select an Action page. Select **Delete Last Collection Item.**
- To add back a deleted collection element, click the element in the document tree to open the Select an Action page. Select **Append Collection Item.**

- To clear all test actions, exit the Document Tester and reopen it.

Chapter 16

Updating Document Schema Target Locations

Understanding Updating Document Schema Target Locations

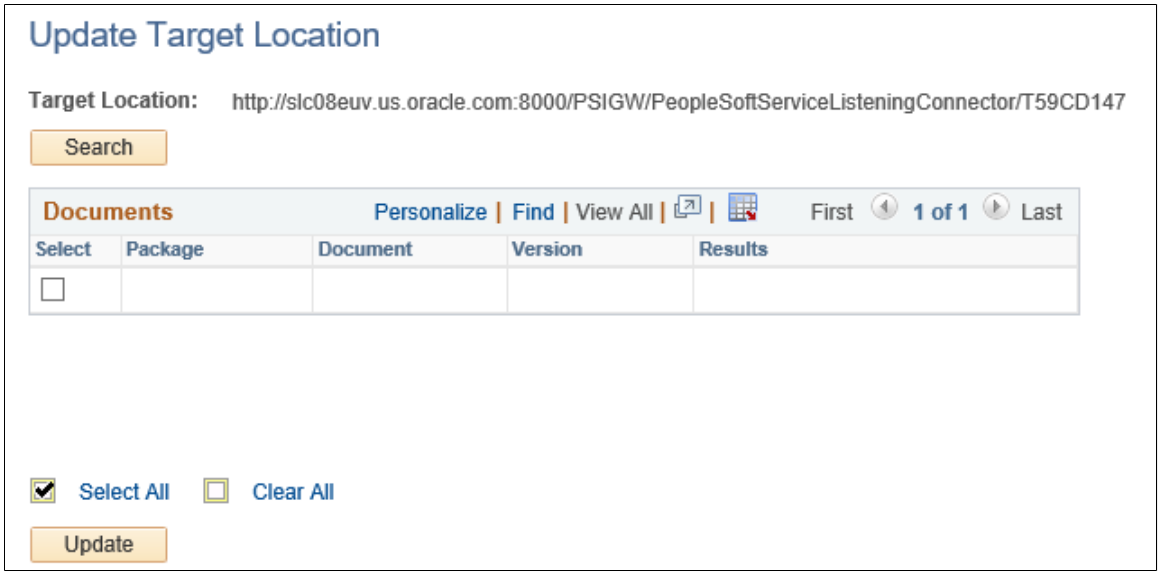
When you save a document, the PeopleSoft system uses the target location URL defined in the Service Configuration page (**PeopleTools** > **Integration Broker** > **Service Configuration**) to build the XML document schema.

The target location for schemas can potentially change in development mode or when documents are imported using the Project Copy process. PeopleSoft provides a Update Target Location utility that enables you to update document schema in the system that are using target locations other than that defined in the Service Configuration page.

Accessing the Update Doc Target Location Utility

To access the Update Doc Target Location utility page (IB_DOCUMENT_TARGET) in the IB_DOCUMENT_TARGET select **PeopleTools** > **Documents** > **Document Utilities** > **Update Doc Target Location.**

This example illustrates the fields and controls on the Update Doc Target Location page.



Updating Target Locations for Document Schema

The Update Doc Target Location utility page lists documents that have a schema defined in a location other than the target location specified on the Service Configuration page. You can rebuild the schemas using the target location specified in the Service Configuration page.

Note that when you access the Update Doc Target Location page, the target location URL as defined on the Service Configuration page appears at the top of the page.

To update schema target locations:

1. Access the Update Doc Target Location page (**PeopleTools** > **Documents** > **Document Utilities** > **Update Doc Target Location**).

2. Click the **Search** button.

   Any schemas that have a target location other than the one defined on the Service Configuration page appear in the **Documents** grid.

3. Select the **Select** check box next to each document to update the schema target location.

4. Click the **Update** button.

Chapter 17

Validating Document References to Object Metadata

Understanding Validating Document References to Metadata

PeopleSoft provides a Validate Document Metadata utility that enables you to identify documents that are not properly coupled to other metadata objects and data. Typically, this situation occurs:

- After the Project Copy process, when related metadata objects are not included in a project.
- During development, when other developers modify metadata, such as records and fields.

After you identify the invalid references, you can either manually correct or create the necessary metadata in the database, or you can use the utility option to clear the invalid reference.

The Validate Document Metadata utility features Documents, Messages, and Relational tabs that enable you to validate metadata references as described in the following table:

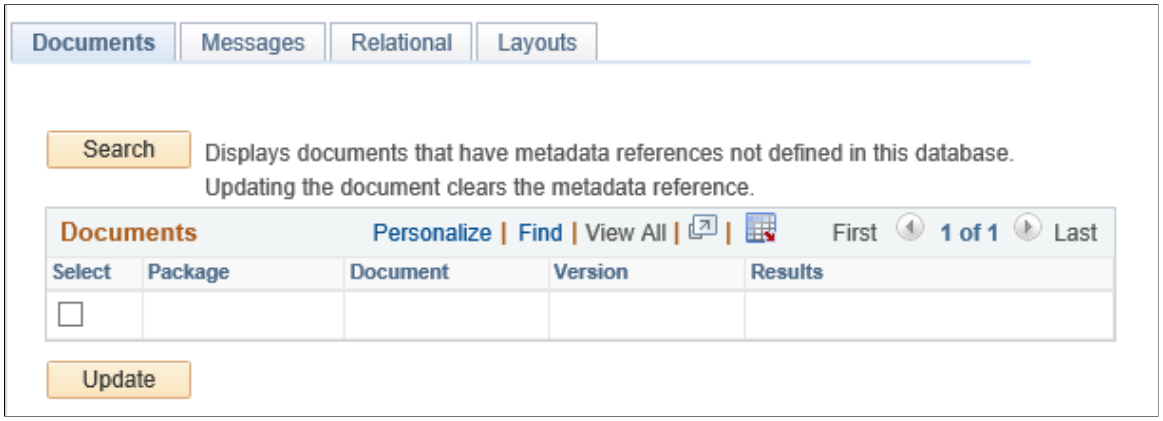
Tab	*Description*
Documents	Displays any documents that have references to metadata that is not in the current database. For an example, *Document A* could contain a compound element that is another document, *Document B*. If *Document B* is not in the current database, then the utility would identify it and display it in the utility.
Messages	You can define documents as a message type in PeopleSoft Integration Broker. The Messages tab displays messages that reference documents that are not defined in this database.
Relational	Displays any invalid record and field maps defined for relational-formatted documents.

Accessing the Validate Document Metadata Utility

The Validate Document Metadata utility (IB_DOCUMENT_VALIDT) is located in the IB_DOCUMENT_VALIDT component.

To access the page, select **PeopleTools** > **Documents** > **Document Utilities** > **Validate Document Metadata .**

This example illustrates the fields and controls on the Validate Document Metadata page.



Validating Document Metadata References

Use the Documents page (IB_DOCUMENT_VALIDT) of the Document/Metadata Validation utility to identify any documents that reference metadata not in the database.

To validate document metadata references:

1. Access the Document/Metadata Validation - Documents page (**PeopleTools** > **Documents** > **Document Utilities** > **Validation Document Metadata**).

2. Click the **Search** button.

   Any documents in the database that reference metadata not in the database appear in the **Documents** grid.

3. Manage the results by performing any of the following actions:

   - Clear the metadata reference.

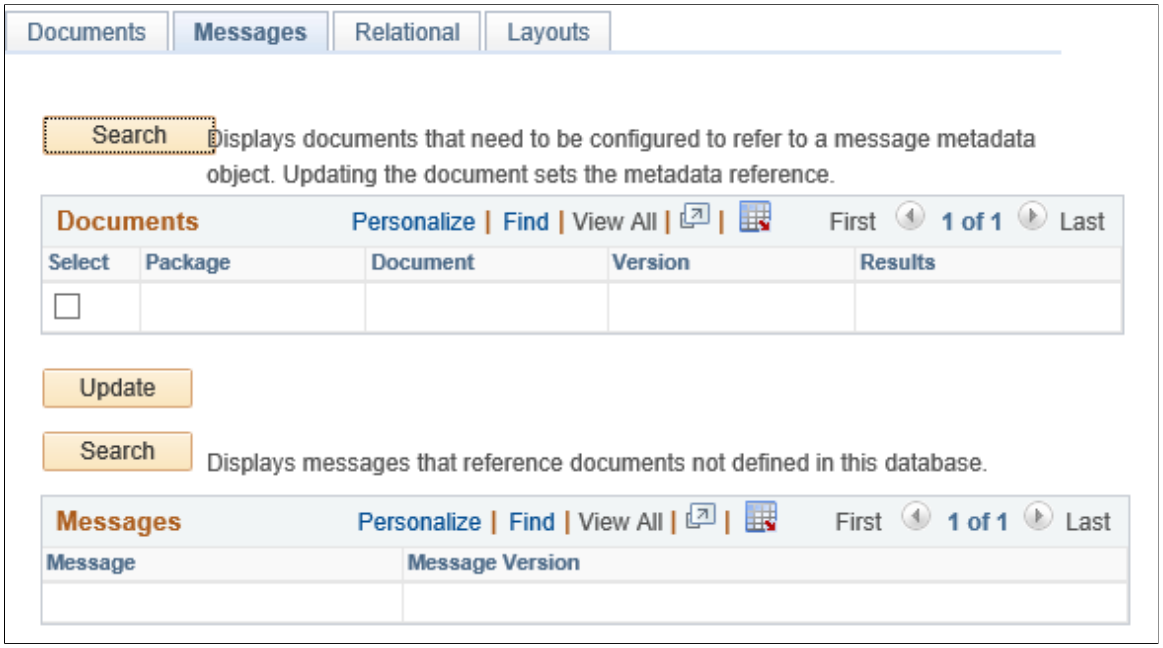
     Select the **Select** check box next to each document for which to clear the invalid or missing reference, and then click the **Update** button.

   - Examine the document definition and import or create the missing metadata.

Validating Document Message Type References

Documents can be defined as a message type in PeopleSoft Integration Broker. Use the Validate Document Metadata - Messages page (IB_DOCUMENT_VDMSG) to resolve issues with the documents or messages used in this message type.

This example illustrates the fields and controls on the Validate Document Metadata - Messages page.



Identifying and Resolving Message Definitions that have Missing References to Documents

A message definition can have a missing reference to a document when the following situation occurs:

- You define a document as a message type in System A. You then copy only the message to System B because System B already has a copy of the document in the database.
- You define a document as a message type in System A. You then copy the message and document to System B in stages, first copying the message definition and then copying the document.

In either case, the message-document link gets broken, and the link needs to be reestablished. Use the upper grid on the Validate Document Metadata - Messages page to identify and resolve this inconsistency.

To identify and resolve message definitions that have missing references to documents:

1. Access the Validate Document Metadata - Document page (**PeopleTools** > **Documents** > **Document Utilities** > **Validate Document Metadata.**).
2. Click the **Messages** tab.

   The Validate Document Metadata - Messages page appears.
3. Above the **Documents** grid, click the **Search** button to display all documents in the system that have missing references to message definitions.

   The results appear in the **Documents** grid.
4. Select the **Select** box next to each document you want to update.
5. Click the **Update** button.

Identifying Messages that Reference Documents Not in the Current Database

Use the bottom grid on the Validate Document Metadata - Messages page to identify messages in the system that reference documents that are not currently defined in the database.

This situation can occur when you define a document as a message type in PeopleSoft Integration Broker in a database, and then copy the message to a new database and do not include the document in the project copy process.

To identify messages with invalid references to documents:

1. Access the Validate Document Metadata- Documents page (**PeopleTools** > **Documents** > **Document Utilities** > **Validate Document Metadata**).
2. Click the **Messages** tab.

   The Validate Document Metadata - Messages page appears.
3. Just above the **Messages** grid at the bottom of the page, click the **Search** button.

   Any messages in the database that reference documents not in the database appear in the **Messages** grid.

To resolve the invalid references, go to the source database and copy the documents to the current database or create new documents in the current database.

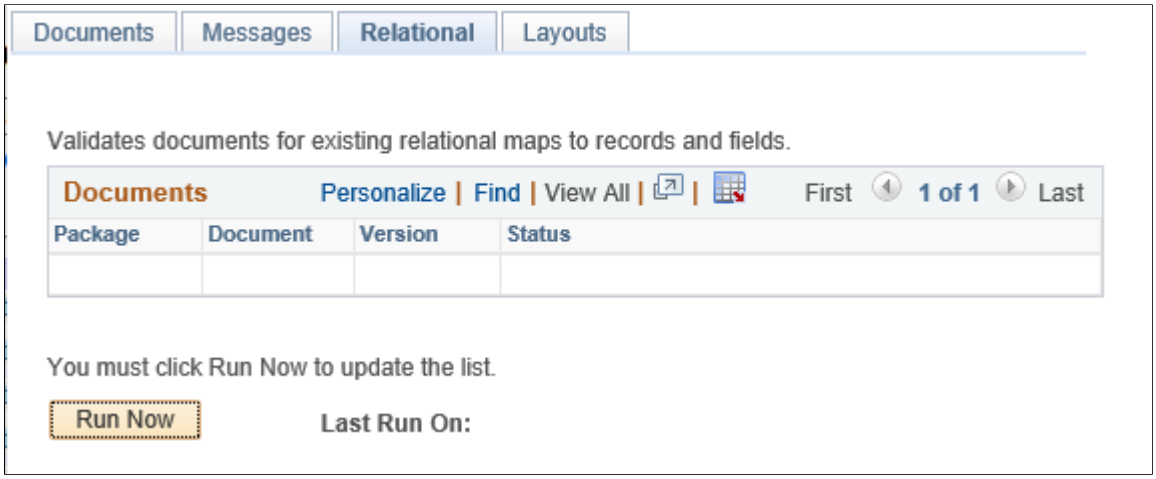
Validating Relational Document Record and Field Maps

For documents that have relational maps defined to PeopleSoft records and fields, other database users may be able to modify the records or fields. When changes are made to records and fields, PeopleSoft Application Designer does not perform any validation against document metadata.

Use the Validate Document Metadata - Relational page (IB_DOCUMENT_VDREC) to run an application engine program to validate all documents in the database that have relational maps defined.

To access the Document/Metadata Validation - Relational page, select **PeopleTools** > **Documents** > **Document Utilities** > **Validate Document Metadata** and click the **Relational** tab.

This example illustrates the fields and controls on the Validate Document Metadata - Relational.



If the application engine program was run previously on the database, the date and time it was run appear in **Last Run On** field.

To validate relational document record and field maps:

1. Access the Validate Document Metadata - Relational page (**PeopleTools** > **Documents** > **Document Utilities** > **Validate Document Metadata** and click the **Relational** tab).

2. Click the **Run Now** button.

Documents that have invalid relational maps appear in the **Documents** grid. Open each document and correct the relational map.

