ORACLE
JD EDWARDS ENTERPRISEONE

JD Edwards EnterpriseOne Business Services 9.0 Reference Guide

September 2008

ORACLE

JD Edwards EnterpriseOne Business Services 9.0 Reference Guide
SKU e190AFI-B 0908

Trademark Notice

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

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Subject to patent protection under one or more of the following U.S. patents: 5,781,908; 5,828,376; 5,950,010; 5,960,204; 5,987,497; 5,995,972; 5,987,497; and 6,223,345. Other patents pending.

Warranty Disclaimer

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

Restricted Rights Notice

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

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

Third Party Content, Products, and Services Disclaimer

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

Open Source Disclosure

Oracle takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation. The following open source software may be used in Oracle's JD Edwards EnterpriseOne products and the following disclaimers are provided:

Contents

About This Documentation Preface

JD Edwards EnterpriseOne implementation guides provide you with the information that you need to implement and use JD Edwards EnterpriseOne applications from Oracle.

This preface discusses:

- JD Edwards EnterpriseOne application prerequisites.
- Application fundamentals.
- Documentation updates and downloading documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common fields in implementation guides.

Note. Implementation guides document only elements, such as fields and check boxes, that require additional explanation. If an element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common fields for the section, chapter, implementation guide, or product line. Fields that are common to all JD Edwards EnterpriseOne applications are defined in this preface.

JD Edwards EnterpriseOne Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use JD Edwards EnterpriseOne applications.

You might also want to complete at least one introductory training course, if applicable.

You should be familiar with navigating the system and adding, updating, and deleting information by using JD Edwards EnterpriseOne menus, forms, or windows. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your JD Edwards EnterpriseOne applications most effectively.

Application Fundamentals

Each application implementation guide provides implementation and processing information for your JD Edwards EnterpriseOne applications.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals implementation guide. Most product lines have a version of the application fundamentals implementation guide. The preface of each implementation guide identifies the application fundamentals implementation guides that are associated with that implementation guide.

The application fundamentals implementation guide consists of important topics that apply to many or all JD Edwards EnterpriseOne applications. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals implementation guides. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Downloading Documentation

This section discusses how to:

- Obtain documentation updates.
- Download documentation.

Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on Oracle's PeopleSoft Customer Connection website. Through the Documentation section of Oracle's PeopleSoft Customer Connection, you can download files to add to your Implementation Guides Library. You'll find a variety of useful and timely materials, including updates to the full line of JD Edwards EnterpriseOne documentation that is delivered on your implementation guides CD-ROM.

Important! Before you upgrade, you must check Oracle's PeopleSoft Customer Connection for updates to the upgrade instructions. Oracle continually posts updates as the upgrade process is refined.

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Downloading Documentation

In addition to the complete line of documentation that is delivered on your implementation guide CD-ROM, Oracle makes JD Edwards EnterpriseOne documentation available to you via Oracle's website. You can download PDF versions of JD Edwards EnterpriseOne documentation online via the Oracle Technology Network. Oracle makes these PDF files available online for each major release shortly after the software is shipped.

See Oracle Technology Network, http://www.oracle.com/technology/documentation/psftent.html

Additional Resources

The following resources are located on Oracle's PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps

Resource	Navigation
Interactive Services Repository	Support, Documentation, Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Pre-Built Integrations for PeopleSoft Enterprise and JD Edwards EnterpriseOne Applications
Minimum technical requirements (MTRs)	Implement, Optimize + Upgrade; Implementation Guide; Supported Platforms
Documentation updates	Support, Documentation, Documentation Updates
Implementation guides support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Release Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in implementation guides:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Indicates field values, emphasis, and JD Edwards EnterpriseOne or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter *O*.
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press the W key.
`Monospace font`	Indicates a PeopleCode program or other code example.
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
. . . (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.

Visual Cues

Implementation guides contain the following visual cues.

Notes

Notes indicate information that you should pay particular attention to as you work with the JD Edwards EnterpriseOne system.

Note. Example of a note.

If the note is preceded by *Important!,* the note is crucial and includes information that concerns what you must do for the system to function properly.

Important! Example of an important note.

Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

Warning! Example of a warning.

Cross-References

Implementation guides provide cross-references either under the heading "See Also" or on a separate line preceded by the word *See.* Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, 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 country-specific heading: "(FRA) Hiring an Employee"

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

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in implementation guides:

- 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 implementation guides:

- USF (U.S. Federal)

- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about implementation guides and other Oracle reference and training materials. Please send your suggestions to your product line documentation manager at Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. Or email us at appsdoc@us.oracle.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

Common Fields Used in Implementation Guides

Address Book Number

Enter a unique number that identifies the master record for the entity. An address book number can be the identifier for a customer, supplier, company, employee, applicant, participant, tenant, location, and so on. Depending on the application, the field on the form might refer to the address book number as the customer number, supplier number, or company number, employee or applicant ID, participant number, and so on.

As If Currency Code

Enter the three-character code to specify the currency that you want to use to view transaction amounts. This code enables you to view the transaction amounts as if they were entered in the specified currency rather than the foreign or domestic currency that was used when the transaction was originally entered.

Batch Number

Displays a number that identifies a group of transactions to be processed by the system. On entry forms, you can assign the batch number or the system can assign it through the Next Numbers program (P0002).

Batch Date

Enter the date in which a batch is created. If you leave this field blank, the system supplies the system date as the batch date.

Batch Status

Displays a code from user-defined code (UDC) table 98/IC that indicates the posting status of a batch. Values are:

Blank: Batch is unposted and pending approval.

A: The batch is approved for posting, has no errors and is in balance, but has not yet been posted.

D: The batch posted successfully.

E: The batch is in error. You must correct the batch before it can post.

P: The system is in the process of posting the batch. The batch is unavailable until the posting process is complete. If errors occur during the post, the batch status changes to *E*.

U: The batch is temporarily unavailable because someone is working with it, or the batch appears to be in use because a power failure occurred while the batch was open.

Branch/Plant

Enter a code that identifies a separate entity as a warehouse location, job, project, work center, branch, or plant in which distribution and manufacturing activities occur. In some systems, this is called a business unit.

Business Unit

Enter the alphanumeric code that identifies a separate entity within a business for which you want to track costs. In some systems, this is called a branch/plant.

Category Code

Enter the code that represents a specific category code. Category codes are user-defined codes that you customize to handle the tracking and reporting requirements of your organization.

Company

Enter a code that identifies a specific organization, fund, or other reporting entity. The company code must already exist in the F0010 table and must identify a reporting entity that has a complete balance sheet.

Currency Code

Enter the three-character code that represents the currency of the transaction. JD Edwards EnterpriseOne provides currency codes that are recognized by the International Organization for Standardization (ISO). The system stores currency codes in the F0013 table.

Document Company

Enter the company number associated with the document. This number, used in conjunction with the document number, document type, and general ledger date, uniquely identifies an original document.

If you assign next numbers by company and fiscal year, the system uses the document company to retrieve the correct next number for that company.

If two or more original documents have the same document number and document type, you can use the document company to display the document that you want.

Document Number

Displays a number that identifies the original document, which can be a voucher, invoice, journal entry, or time sheet, and so on. On entry forms, you can assign the original document number or the system can assign it through the Next Numbers program.

Document Type

Enter the two-character UDC, from UDC table 00/DT, that identifies the origin and purpose of the transaction, such as a voucher, invoice, journal entry, or time sheet. JD Edwards EnterpriseOne reserves these prefixes for the document types indicated:

P: Accounts payable documents.

R: Accounts receivable documents.

T: Time and pay documents.

I: Inventory documents.

O: Purchase order documents.

S: Sales order documents.

Effective Date	Enter the date on which an address, item, transaction, or record becomes active. The meaning of this field differs, depending on the program. For example, the effective date can represent any of these dates: • The date on which a change of address becomes effective. • The date on which a lease becomes effective. • The date on which a price becomes effective. • The date on which the currency exchange rate becomes effective. • The date on which a tax rate becomes effective.
Fiscal Period and **Fiscal Year**	Enter a number that identifies the general ledger period and year. For many programs, you can leave these fields blank to use the current fiscal period and year defined in the Company Names & Number program (P0010).
G/L Date (general ledger date)	Enter the date that identifies the financial period to which a transaction will be posted. The system compares the date that you enter on the transaction to the fiscal date pattern assigned to the company to retrieve the appropriate fiscal period number and year, as well as to perform date validations.

JD Edwards EnterpriseOne Business Services Preface

This preface discusses:

- JD Edwards EnterpriseOne products.
- Additional sources of information.
- Common terms used in this guide.

JD Edwards EnterpriseOne Products

This implementation guide refers to these JD Edwards EnterpriseOne products from Oracle:

- JD Edwards EnterpriseOne Accounts Payable
- JD Edwards EnterpriseOne Accounts Receivable
- JD Edwards EnterpriseOne Address Book
- JD Edwards EnterpriseOne Capital Asset Management
- JD Edwards EnterpriseOne Customer Relationship Management
- JD Edwards EnterpriseOne Expense Management
- JD Edwards EnterpriseOne Fixed Asset Management
- JD Edwards EnterpriseOne General Accounting
- JD Edwards EnterpriseOne Inventory Management
- JD Edwards EnterpriseOne Procurement
- JD Edwards EnterpriseOne Sales Order Management
- JD Edwards EnterpriseOne Service Management

Customers must conform to the supported platforms for the release as detailed in the JD Edwards EnterpriseOne minimum technical requirements. In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the cross-reference material in the Program Documentation at http://oracle.com/contracts/index.html for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

Additional Sources of Information

Before you can run JD Edwards EnterpriseOne business services, you must be sure that the technical setup of your system and of your applications is correct. Additional, essential information about the technical system setup and application setup that is required to run JD Edwards EnterpriseOne business services appears in:

- JD Edwards EnterpriseOne 8.98 tools guides.
- JD Edwards EnterpriseOne 9.0 application implementation guides.
- Technical documents and presentations about business services and service-oriented architecture (SOA).
- JD Edwards EnterpriseOne 8.98 Business Services Documentation Map.
- JD Edwards EnterpriseOne 8.98 Reference Implementations Guide.

The steps that you need to take to configure your system are outlined in the JD Edwards EnterpriseOne 8.98 Business Services Documentation Map. This document includes each step that a user might need to complete, along with a reference to the documentation that describes each step. You should review all of the information that is included in the documentation map before attempting to set up and configure your system to use business services.

Reference implementations are complete beginning-to-end examples of implemented business services. Reference implementation business services are fully functional and can be used to test your business service environment.

Note. Standard tools and application guides are located on the documentation CD that accompanies your software. You can access all of the additional sources of information, including the technical presentations, the documentation map, reference implementation information, mapping spreadsheets, and additional documentation on Oracle's Customer Connection web site. From the main Customer Connection page, select Implement, Optimize and Upgrade. Then select Implementation Guide and then click on the Implementation Documentation and Software link. Scroll down the page and click on the Service Oriented Architecture (SOA) link.

Customers must conform to the supported platforms for the release as detailed in the JD Edwards EnterpriseOne minimum technical requirements. In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the cross-reference material in the Program Documentation at http://oracle.com/contracts/index.html for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

See Also

JD Edwards EnterpriseOne 8.98 Business Services Documentation Map

JD Edwards EnterpriseOne 8.98 Business Services Development Guide

JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide

JD Edwards EnterpriseOne 8.98 Reference Implementations Guide

JD Edwards EnterpriseOne 8.98 Business Services Server Reference Guide

JD Edwards EnterpriseOne 8.98 Tools Interoperability Guide

JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide, "JD Edwards EnterpriseOne Accounts Payable Preface"

JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide, "JD Edwards EnterpriseOne Accounts Receivable Preface"

JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide, "JD Edwards EnterpriseOne Address Book Preface"

JD Edwards EnterpriseOne Capital Asset Management 9.0 Implementation Guide, "JD Edwards EnterpriseOne Capital Asset Management Preface"

JD Edwards EnterpriseOne Customer Relationship Management Application Fundamentals 9.0 Implementation Guide, "JD Edwards EnterpriseOne CRM Application Fundamentals Preface"

JD Edwards EnterpriseOne Financial Management Application Fundamentals 9.0 Implementation Guide, "JD Edwards EnterpriseOne Financial Management Application Fundamentals Preface"

JD Edwards EnterpriseOne Fixed Assets 9.0 Implementation Guide, "JD Edwards EnterpriseOne Fixed Assets Preface"

JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide, "JD Edwards EnterpriseOne Inventory Management Preface"

JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide, "JD Edwards EnterpriseOne Procurement Management Preface"

JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide, "JD Edwards EnterpriseOne Sales Order Management Preface"

Common Terms Used in this Guide

BPEL PM (Business Process Execution Language Process Manager	A plug-and-play, standard-based infrastructure for integrating systems, services and activities into process flows that are easy to change. BPEL PM can deliver both composite applications, such as web service orchestration or J2EE process flows, and data integration applications such as file-to-DB, DB-to-DB, or OracleApps-to-X.
Business Service	Business services are JD Edwards EnterpriseOne OMW objects that contains one or more java classes that expose public methods. The methods access logic in EnterpriseOne and support a specific step in a business process, such as adding a sales order or updating an address book record.

A business service contains internal value object classes, which make up the signature for the public methods.

Business services are managed by published business services, which are exposed to the user as a web service.

Note. In the JD Edwards EnterpriseOne system, business services are managed by published business services. These published business services are exposed to the user as web services. Therefore, in the context of web services, business services are called web service operations. This documentation uses the terms business service and web service operation interchangeably.

Business Service Property

Business service properties are key value pairs that are referenced in the business service code. The properties enable users to change the functionality of a business service without changing the internal code of the business service. Properties also enable users to expose values to an end user of the business service.

For example, a customer can use business service properties to specify a particular version of a program to use when processing data that is associated with the business service. Business service properties enable users to easily enter or change the version without having to change the code of the associated business service.

Class

See *Java Class*.

Component

Components are the extensible building blocks of a value object, and are comprised of compounds and fields, or fields alone.

Compounds

Compounds are collections of related fields within a value object. An example of a compound is Entity, which is made up of fields entityID, entityLongId, and entityTaxId.

ESB (Enterprise Service Bus)

A component of the BPEL PM product. An ESB provides integration infrastructure that combines SOA and event-driven architecture (EDA) functionality to enable organizations to respond to business changes quickly and easily. The integration of ESB with enterprise applications, BPEL PM, BAM, and web services managers enables organizations to leverage their existing technology investments.

Fields

Fields are the lowest-level elements in a value object. Fields store business data.

Interface

An interface defines the communication boundary between the consumer and provider of a business service. The interface is the consumer's only exposure to the business service. It exposes the operation name and the defined messages that are sent back and forth. Both input and output data associated with a business service is considered part of the interface.

Java Class

A java class is a programming construct that is used to group related variables and methods. The class determines the type of object. For example, AddressBookProcessor.java is a type of business service which is a class.

Javadoc

Javadoc is a computer software tool that enables users to generate application programming interface (API) documentation in HTML format from Java source code. Javadoc is the industry standard for documenting Java classes.

Method

Methods are pieces of code that perform a specific function. A public method is available to be exposed. Some public methods are exposed as web service operations. Other public methods are not exposed, and are internal business services.

Payload

The payload is the business data that is sent by a web service. The payload for an EnterpriseOne business service is the value object.

Published Business Service

A published business service is a JD Edwards EnterpriseOne OMW object that contains or consists of one Java class, which publishes multiple methods. The published business service gives exposure to one or more business services by providing an interface that is available to the public as a consumable web service. Published business services contain value objects that contain data that is passed in and out of the related business services.

Note. In the context of JD Edwards EnterpriseOne, published business services are exposed to consumers as web services. Therefore, in many cases, you will see the terms business service and web service used interchangeably.

See *Business Services*

SOA (Service Oriented Architecture)

A technical architecture that enables organizations to extend the use of their business data beyond their native software systems. SOA enables users to expose native EnterpriseOne business services through web service standards.

Transport

The communication method that a business service uses. The transport defines how and where data is sent.

Value Object

Value objects are the payload, or input and output data, for a business service. Value objects provide the interface for a published business service or business service. Published business services use published value objects and business services use internal value objects. Value objects can contain both business data and warning messages that are produced during data processing. Value objects are comprised of components, compounds and fields.

Web Service

Web services enable software applications that are written in various programming languages and are running on various platforms to exchange data over computer networks. A web service is prescribed by its interface. The public interface of a specific web service is described by Web Services Description Language (WSDL). Web service communication consists of transport and payload.

Early web service implementation used HTTP as a transport and SOAP to describe the payload. Current web service standards have adopted Web Service Invocation Framework (WSIF), allowing SOAP/HTTP, java, and other transport and payload types.

Note. In the context of JD Edwards EnterpriseOne, published business services are exposed as web services. Therefore, in many cases, you will see the terms used interchangeably.

See *Published Business Service*

Web Service Operation

See *Business Service.*

WSDL (Web Service Description Language)	An XML format for describing network services. WSDL describes the public interface of a web service.

CHAPTER 1

Getting Started with JD Edwards EnterpriseOne Business Services

This chapter discusses:

- JD Edwards EnterpriseOne business services overview.
- JD Edwards EnterpriseOne business services integration process.
- JD Edwards EnterpriseOne business service solutions.
- JD Edwards EnterpriseOne business services implementation.

Understanding JD Edwards EnterpriseOne Business Services

Business services are objects that enable interoperability between JD Edwards EnterpriseOne and other Oracle applications or third-party applications and systems. Business services enable software applications that are written in various programming languages and running on various platforms to exchange information.

A web service is a standardized way of integrating web-based applications. In JD Edwards EnterpriseOne, published business services are exposed to consumers as web services. These web services enable JD Edwards EnterpriseOne to expose native transactions as a basic service that can expose an XML document-based interface.

Note. In the JD Edwards EnterpriseOne system, published business services are exposed as web services. Therefore, in many cases, you will see the terms used interchangeably.

Published Business Services

A published business service is a JD Edwards EnterpriseOne object that is managed in Object Management Workbench (OMW). The published business service represents one Java class that publishes multiple business services. When you create a business service or web service, you identify the Java class. The published business service also contains value object classes that make up the signature for the published business service.

Business Services

A business service is a JD Edwards EnterpriseOne object that is managed by OMW. A business service represents one or more classes that expose public methods. Each method performs a business process. A business service also contains internal value object classes that make up the signature for the business service methods. These public methods can be called from other business service classes and published business service classes. Some business services are exposed to consumers, while others are internal. Business services are created for internal use by JD Edwards EnterpriseOne published business services. They are made available to consumers as a call from a published business service. Business services that are exposed to consumers are exposed as web service operations.

Note. In the context of EnterpriseOne, business services are exposed as web service operations. Therefore, in many cases, you will see the term used interchangeably.

Business Service Properties

Many business services use business services properties. Business service properties are similar to processing options for an interactive or batch program in the JD Edwards EnterpriseOne system. Properties enable you to easily change values or processing information about a business service without having to modify the programming code associated with that business service.

For example, many business services include a business service property that enables the user to specify the maximum number of records to return for a query. Other business services include properties that enable users to specify the version of a specific JD Edwards EnterpriseOne program to use during processing.

This documentation includes details about all of the business service properties that are used by each business service.

Utilities

Utilities are generic, reusable business services that perform standard operations that are used by many business services.

Utilities enable multiple business services to complete the same process in a uniform manner. For example, both the ProcessPurchaseOrder and ProcessCustomer business services retrieve entity information from the JD Edwards EnterpriseOne system. Rather than coding the retrieval of entity information separately in both of the business services, each business service uses the Entity Processor utility to retrieve the entity information. By using the Entity Processor utility, the complexity and the amount of code associated with each of the business services is reduced.

Business services use these utilities:

- Entity Processor (J0100010)
- GL Account Processor (J0900010)
- Inventory Item ID Processor (J4100010)
- Net Change Processor (J0000020)
- Processing Version Processor (J0000010)

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide*

JD Edwards EnterpriseOne Business Services Integration Process

Published business services (web services) use business services (web service operations) to transfer information between JD Edwards EnterpriseOne and other Oracle or third-party systems. Because these systems may be using different programming languages or running on different platforms, data must be formatted during transfer so that it can be read and processed by each system.

The user of a third-party system, also known as the consumer, initiates a business services integration by entering data into an input interface. An input interface is based on JD Edwards EnterpriseOne data structure, and is used to hold the information associated with the user's request. For example, the consumer might specify that they want to inquire on a particular sales order in the JD Edwards EnterpriseOne system. Therefore, they can enter the sales order number in the input interface and specify that they want to inquire on that order.

The published business service reads the consumer's request to determine the action that the system takes, and calls the appropriate business service. In this example the SalesOrderManager published business service calls the GetSalesOrder business service. Additionally, the published business service formats the data in the input interface so that it can be read and processed by the JD Edwards EnterpriseOne system.

The business service passes the data into the EnterpriseOne system and tells the system what actions to perform. After the EnterpriseOne system processes or retrieves the specified information, it sends the results back to the business service. The business service receives the processed data from the EnterpriseOne system and passes it back to the published business service. The published business service then formats the data so that it can be read by the third-party system and passes it back to the consumer using the response interface.

JD Edwards EnterpriseOne Business Service Solutions

Oracle provides several predefined published business services and business services.

AccountsPayableManager

The AccountsPayableManager published business service manages the execution of these business services:

- getVoucher
- processVoucher

See Chapter 3, "Accounts Payable Manager Web Service," page 33.

AddressBookManager

The AddressBookManager published business service manages the execution of these business services:

- getAddressBook
- processAddressBook
- getContact
- processContact

See Chapter 4, "Address Book Manager Web Service," page 39.

CapitalAssetManager

The CapitalAssetManager published business service manages the execution of these business services:

- createCapitalAssetConditionBasedAlert
- getCapitalAssetConditionBasedAlert

See Chapter 5, "Capital Asset Manager Web Service," page 49.

CustomerManager

The CustomerManager published business service manages the execution of these business services:

- getCustomer
- getCustomerCreditInformation
- processCustomer

See Chapter 6, "Customer Manager Web Service," page 57.

CustomerServiceManager

The CustomerServiceManager published business service manages the execution of these business services:

- getServiceOrder
- processServiceOrder
- getCommitmentDateTime

See Chapter 7, "Customer Service Manager Web Service," page 65.

EquipmentManager

The EquipmentManager published business service manages the execution of these business services:

- equipmentProcessor
- equipmentQueryProcessor

See Chapter 8, "Equipment Manager Web Service," page 75.

FinancialComplianceManager

The FinancialComplianceManager published business service manages the execution of these business services:

- getAPProcessingOptions
- getWriteOffProcessing Options
- getAgingCompanyConstants
- getCustomerCreditLimits
- getGeneralConstants
- getJournalGeneralConstants
- getPolicyEditRules
- getPurchasingToleranceRules
- getSecurityWorkbench

See Chapter 9, "Financial Compliance Manager Web Service," page 81.

FinancialsManager

The FinancialsManager published business service manages the execution of these business services:

- getGLAccount
- insertBatchJournalEntry

See Chapter 10, "Financials Manager Web Service," page 95.

FixedAssetManager

The FixedAssetManager published business service manages the execution of the getFixedAsset business service.

See Chapter 11, "Fixed Asset Manager Web Service," page 99.

FoundationEnvironmentManager

The FoundationEnvironmentManager published business service manages the execution of the getUserDefinedCode business service.

See Chapter 12, "Foundation Environment Web Service," page 103.

InventoryManager

The InventoryManager published business service manages the execution of these business services:

- processSupplierCatalogPrice
- processInventoryItem
- getBranchPlantItem
- getSupplierCatalogPrice
- getItemAvailability
- insertInventoryItemStaging

See Chapter 13, "Inventory Manager Web Service," page 107.

ProcurementManager

The ProcurementManager published business service manages the execution of these business services:

- processPurchaseOrder
- processPurchaseOrderAcknowledge
- getPurchaseOrder

See Chapter 14, "Procurement Manager Web Service," page 121.

PurchaseOrderReceiptManager

The PurchaseOrderReceiptManager published business service manages the execution of the processPurchaseOrderReceipt business service.

See Chapter 15, "Purchase Order Receipt Manager Web Service," page 133.

SalesOrderManager

The SalesOrderManager published business service manages the execution of these business services:

- processSalesOrder
- processSalesPriceAdjustment
- getItemPriceAndAvailability
- getCustomerItemPrice
- getSalesOrder

- getItemListPrice
- getSalesOrderPriceHistory

See Chapter 16, "Sales Order Manager Web Service," page 139.

SupplierManager

The SupplierManager published business service manages the execution of these business services:

- processSupplier
- getSupplier

See Chapter 17, "Supplier Manager Web Service," page 163.

VoucherMatchManager

The VoucherMatchManager published business service manages the execution of the processVoucherMatch business service.

See Chapter 18, "Voucher Match Manager Web Service," page 169.

JD Edwards EnterpriseOne Business Services Implementation

Before you can use JD Edwards EnterpriseOne business services, you must be sure that the technical setup of your system supports Service Oriented Architecture (SOA) integration processing. All of the system setup steps that you must perform are outlined in the JD Edwards EnterpriseOne 8.98 Documentation Map, which you can access on the JD Edwards Service Oriented Architecture (SOA) page from Oracle's Customer Connection web site.

In addition to technical setup, you must set up these application systems before you can use the business services that are included in this documentation:

- JD Edwards EnterpriseOne Accounts Payable
- JD Edwards EnterpriseOne Accounts Receivable
- JD Edwards EnterpriseOne Address Book
- JD Edwards EnterpriseOne Customer Relationship Management
- JD Edwards EnterpriseOne Inventory Management
- JD Edwards EnterpriseOne Procurement
- JD Edwards EnterpriseOne Sales Order Management

The implementation steps that are required for each of these systems are documented in detail in the corresponding implementation guide.

CHAPTER 2

Accessing Additional Information about Business Services

This chapter discusses how to

- Access Javadoc for business services.
- Review input and response interface tables.

Accessing Javadoc for Business Services

This section provides an overview of Javadoc for business services and discusses how to:

- Access Javadoc.
- View value object and field information.
- View the uses of an object in Javadoc.
- View hierarchy information.

Understanding Javadoc for Business Services

Javadoc is a tool that parses the declarations and documentation comments in a set of Java source files and produces a corresponding set of HTML pages. These pages describe the public and protected classes, nested classes, interfaces, constructors, methods, and fields. Javadoc generates the Application Programming Interface (API) documentation for the JD Edwards EnterpriseOne business services source files.

Javadoc is installed automatically when you install the JD Edwards EnterpriseOne system. You access Javadoc by navigating to the javadoc folder that is located in your JD Edwards EnterpriseOne install directory. This is an example of the path you would follow to access javadoc:

```
C:\B9\STAGINGA\java\javadoc
```

Note. Javadoc generates several main pages of documentation, each of which can be launched by double clicking on the corresponding html file in the javadoc folder. This documentation provides instructions for accessing and navigating through the API documentation by launching the overview-summary.html file. If you choose to access the documentation by launching a different file, the instructions might be slightly different.

Javadoc is organized into several main pages of documentation, each of which is described in this table:

Page Title	Description
Overview	The overview page is the main page of API documentation. This page contains a list of all of the available packages and a link to each package.
Package	Each package has its own page that contains a list of each class and interface. The page also contains a summary of each class and interface and provides a link to each.
Class	Each class, interface, nested class and nested interface has its own page. Each page contains a brief description of the class/interface. The page also includes summary and detail tables for field, constructor and method information. The page also includes information about inherited methods.
Tree	Each package has a tree or hierarchy page. The information that appears in the Tree page depends on the page from which the Tree option is selected. Clicking the Tree option from the overview page displays the hierarchy for all packages. Clicking the Tree option from a particular package, class or interface page displays the hierarchy for only that package.
Index	The index page contains an alphabetical list of all classes, interfaces, constructors, methods and fields. You access the index by clicking the Index link in the navigation bar from any page.
Help	The help page provides general instructions about how to use javadoc and overview information about how the documentation is organized. You access the help page by clicking the Help link in the navigation bar from any page.

Display and Navigation Options

To navigate through the Javadoc documentation, you can click on any of the package, class, method, value object or field names that are live links. When you click on these links a new page opens with additional detail about that object. You can also use the links in the navigation bar at the top of each page. The standard navigation bar includes these links:

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

All javadoc pages are available with or without HTML frames. To view the page with or without frames, click on one of these options at the top of any page:

- Frames
- No Frames

You can also use the previous and next options at the top of each page to view the next package, class, or interface. Depending on the page you are on, the names of the options change. For example, when you are viewing a package, you can click NEXT PACKAGE or PREV PACKAGE to navigate to another package.

Package Naming Conventions

To access javadoc for a specific business service, you must locate the appropriate package. The naming convention for JD Edwards EnterpriseOne business service packages is:

```
oracle.e1.bssv.<OMW object name>
```

In package naming, objects can be published business services or business services. Published business service object names begin with JP followed by a numeric identification code. For example, the AddressBookManager object name is JP010000. Business service object names begin with *J* followed by a numeric identification code. For example, the processAddressBook business service object name is J0100001. Therefore, to access javadoc for the AddressBookManager business service, you would select one of these packages from the overview page:

- `oracle.e1.bssv.JP010000`

  Selecting this package brings you to the main package page for the AddressBookManager business service.
- `oracle.e1.bssv.JP010000.valueobject`

  Selecting this package brings you directly to a page that lists and describes each value object that is associated with the AddressBookManager business service.

Note. Javadoc for several reference implementations has also been provided. Reference implementation package names include the letter *R* after the *J* or *JP*.

Uses of a Business Service Object

You can also use javadoc to determine where a business service or related object is used. Each documented package, class and interface has its own Use page. The page describes what packages, classes, methods, constructors and fields use any part of the selected class or package. You access the Use page for an object by clicking the Use link in the navigation bar.

For example, if you are viewing class ABC, and you click on Use, the Use page for class ABC displays:

- Subclasses of ABC.
- Fields declared as ABC.
- Methods that return ABC.
- Methods and constructors with parameter type ABC.

Class Hierarchy

You can use Javadoc to display the hierarchy of all packages, or of individual packages, classes or interfaces. You use the Tree link in the navigation bar to access hierarchy information. If you click the Tree link from the main overview page, the system displays a hierarchical view of all packages along with a hierarchical view of each package.

Alternatively, you can click the Tree link from any package, class or interface page to view the hierarchy for only that package.

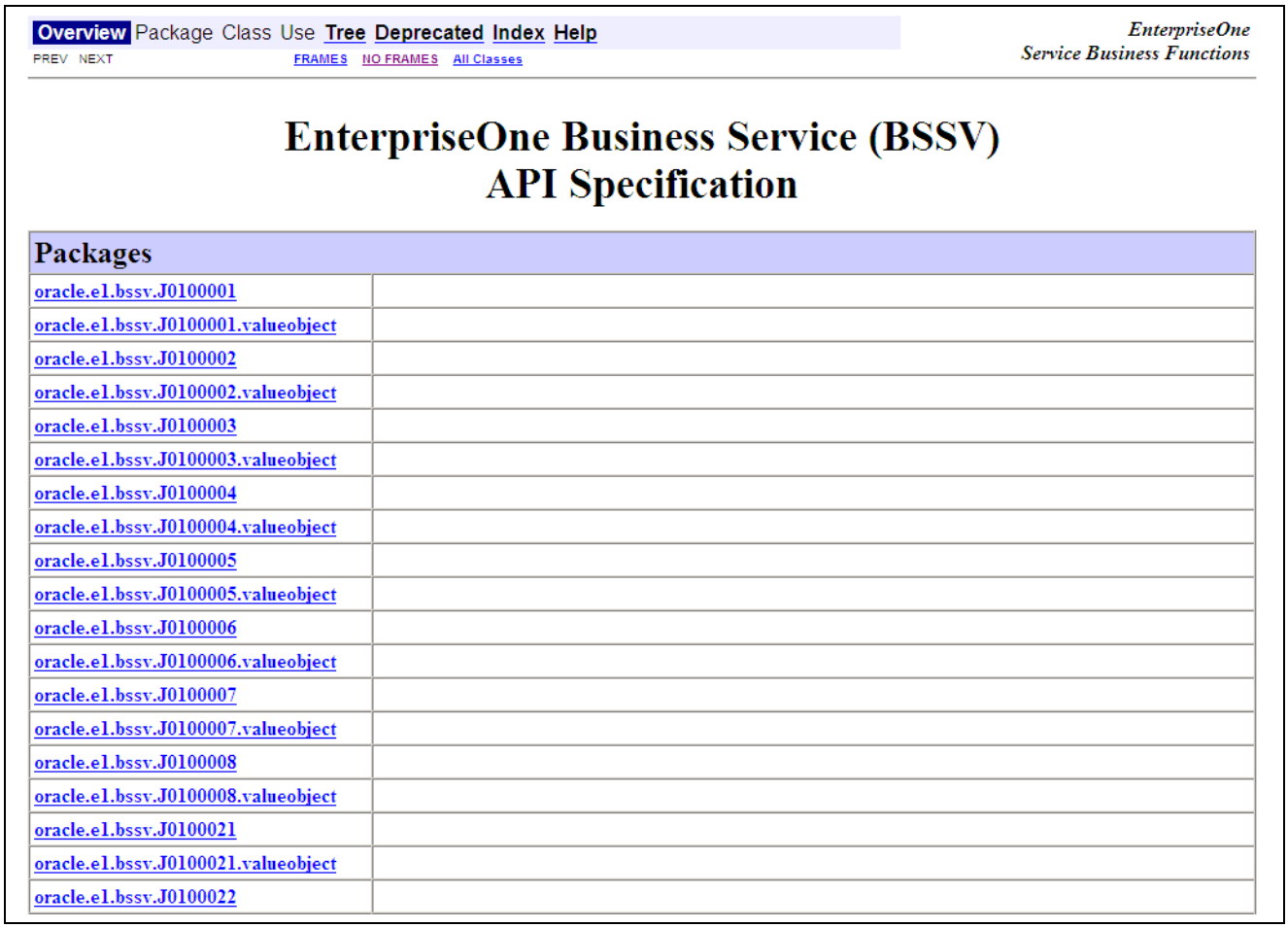
Classes are organized by inheritance structure, starting with `java.lang.object`. The interfaces do not inherit from `java.lang.object`.

Prerequisite

Before you can access Javadoc, you must install your JD Edwards EnterpriseOne system and identify your install directory. Contact your system administrator for assistance.

Accessing Javadoc

Access the Javadoc folder in your JD Edwards EnterpriseOne install directory and double click on the overview-summary.html file.



Overview Package Class Use Tree Deprecated Index Help

PREV NEXT FRAMES NO FRAMES All Classes

EnterpriseOne Service Business Functions

EnterpriseOne Business Service (BSSV) API Specification

Packages	
oracle.e1.bssv.J0100001	
oracle.e1.bssv.J0100001.valueobject	
oracle.e1.bssv.J0100002	
oracle.e1.bssv.J0100002.valueobject	
oracle.e1.bssv.J0100003	
oracle.e1.bssv.J0100003.valueobject	
oracle.e1.bssv.J0100004	
oracle.e1.bssv.J0100004.valueobject	
oracle.e1.bssv.J0100005	
oracle.e1.bssv.J0100005.valueobject	
oracle.e1.bssv.J0100006	
oracle.e1.bssv.J0100006.valueobject	
oracle.e1.bssv.J0100007	
oracle.e1.bssv.J0100007.valueobject	
oracle.e1.bssv.J0100008	
oracle.e1.bssv.J0100008.valueobject	
oracle.e1.bssv.J0100021	
oracle.e1.bssv.J0100021.valueobject	
oracle.e1.bssv.J0100022	

Javadoc Overview page

Your web browser opens the main overview page. To view information about a specific business service, click on the corresponding package name. For example, to view information about the AddressBookManager business service, click on package oracle.e1.bssv.JP010000.

The web browser displays the package page, which includes the Class Summary table for the selected package.

Overview **Package** Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES All Classes

EnterpriseOne Service Business Functions

Package oracle.e1.bssv.JP010000

Interface Summary	
Oracle_E1_SBF_SEI_PkgBldFile_AddressBookManager	

Class Summary	
AddressBookManager	The AddressBook Published Business Service shall manage the execution of the following SBFs: - Process Address Book - Process Contact - Get Address Book - Get Contact

Package oracle.e1.bssv.JP010000 page

Review the information in the Class Summary table and click on the link in the table to view field, constructor and method summary and detail information, along with inherited method information for the business service.

Note. You can access value object information by clicking on the links in the method summary table. For additional information about viewing value objects, proceed to the next task in this section.

Overview Package **Class** Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

EnterpriseOne Service Business Functions

oracle.e1.bssv.JP010000

Class AddressBookManager

```
java.lang.Object
  └oracle.e1.bssvfoundation.base.BusinessService
      └oracle.e1.bssvfoundation.base.PublishedBusinessService
          └oracle.e1.bssv.JP010000.AddressBookManager
```

All Implemented Interfaces:

ServiceLifecycle

```
public class AddressBookManager
extends oracle.e1.bssvfoundation.base.PublishedBusinessService
```

The AddressBook Published Business Service shall manage the execution of the following SBFs: - Process Address Book - Process Contact - Get Address Book - Get Contact

Class AddressBookManager (1 of 7)

Constructor Summary

AddressBookManager() Published Business Service Public Constructor

Method Summary

ShowAddressBook	getAddressBook(GetAddressBook vo) Published method for getAddressBook This exposed method Acts as wrapper method, passing null context and null connection, will call protected Published Business ServiceMethodName.
protected ShowAddressBook	getAddressBook(oracle.e1.bssvfoundation.base.IContext context, oracle.e1.bssvfoundation.connection.IConnection connection, GetAddressBook vo) Protected method for AddressBookManager Published Business Service.
ShowContact	getContact(GetContact vo) Published method for Get Contact Acts as wrapper method, passing null context and null connection, will call protected getContact.
protected ShowContact	getContact(oracle.e1.bssvfoundation.base.IContext context, oracle.e1.bssvfoundation.connection.IConnection connection, GetContact vo) Protected method for extends Published Business Service.
protected ConfirmProcessAddressBook	processAddressBook(oracle.e1.bssvfoundation.base.IContext context, oracle.e1.bssvfoundation.connection.IConnection connection, ProcessAddressBook vo) Protected method for AddressBookManager Published Business Service.
ConfirmProcessAddressBook	processAddressBook(ProcessAddressBook vo) Published method for process address book.
protected ConfirmProcessContact	processContact(oracle.e1.bssvfoundation.base.IContext context, oracle.e1.bssvfoundation.connection.IConnection connection, ProcessContact vo) Protected method for AddressBookManager Published Business Service.
ConfirmProcessContact	processContact(ProcessContact vo) Published method for process contact.

Class AddressBookManager (2 of 7)

Methods inherited from class oracle.e1.bssvfoundation.base.PublishedBusinessService
close, destroy, finishPublishedMethod, init, startPublishedMethod, startPublishedMethod

Methods inherited from class oracle.e1.bssvfoundation.base.BusinessService
finishInternalMethod, startInternalMethod

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

AddressBookManager

```
public AddressBookManager()
```

Published Business Service Public Constructor

Method Detail

processAddressBook

```
public ConfirmProcessAddressBook processAddressBook(ProcessAddressBook vo)
                                 throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Published method for process address book. processAddressBook method shall process adds, changes and deletes of Address Book information in EnterpriseOne Acts as wrapper method, passing null context and null connection, will call protected processAddressBook.

Parameters:
vo - the value object representing input data for
Returns:
confirmVO the response data from the business process
Throws:
oracle.e1.bssvfoundation.exception.BusinessServiceException

Class AddressBookManager (3 of 7)

---

processAddressBook

```
protected ConfirmProcessAddressBook processAddressBook(oracle.e1.bssvfoundation.base.IContext context,
                                                       oracle.e1.bssvfoundation.connection.IConnection connection,
                                                       ProcessAddressBook vo)
                                                 throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Protected method for AddressBookManager Published Business Service. processAddressBook will call make calls to the AddressBookProcessor BSSV class for completing business process.

Parameters:

`vo` - the value object representing input data for

`context` - conditionally provides the connection for the database operation and logging information

`connection` - can either be an explicit connection or null. If null the default connection is used.

Returns:

response value object is the data returned from the business process

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

---

processContact

```
public ConfirmProcessContact processContact(ProcessContact vo)
                                 throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Published method for process contact. The processContact method shall process adds, changes and deletes of Contact information in EnterpriseOne. Acts as wrapper method, passing null context and null connection, will call protected processAddressBook.

Parameters:

`vo` - the value object representing input data for

Returns:

confirmVO the response data from the business process

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

Class AddressBookManager (4 of 7)

processContact

```
protected ConfirmProcessContact processContact(oracle.e1.bssvfoundation.base.IContext context,
                                               oracle.e1.bssvfoundation.connection.IConnection connection,
                                               ProcessContact vo)
                                        throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Protected method for AddressBookManager Published Business Service. processContact will make calls to the ContactProcessor BSSV class for completing business process.

Parameters:

`vo` - the value object representing input data for

`context` - conditionally provides the connection for the database operation and logging information

`connection` - can either be an explicit connection or null. If null the default connection is used.

Returns:

response value object is the data returned from the business process

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

---

getContact

```
public ShowContact getContact(GetContact vo)
                      throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Published method for Get Contact Acts as wrapper method, passing null context and null connection, will call protected getContact.

Parameters:

`vo` - the value object representing input data for

Returns:

confirmVO the response data from the business process

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

Class AddressBookManager (5 of 7)

getContact

```
protected ShowContact getContact(oracle.e1.bssvfoundation.base.IContext context,
                                 oracle.e1.bssvfoundation.connection.IConnection connection,
                                 GetContact vo)
                          throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Protected method for extends Published Business Service. getContact will call make calls to the ContactQueryProcessor BSSV class for querying on AdddressBook info

Parameters:

`vo` - the value object representing input data for querying on AdddressBook info..
`context` - conditionally provides the connection for the database operation and logging information
`connection` - can either be an explicit connection or null. If null the default connection is used.

Returns:

confirmVO the response data from the business process AddressBook query.

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

---

getAddressBook

```
public ShowAddressBook getAddressBook(GetAddressBook vo)
                               throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Published method for getAddressBook This exposed method Acts as wrapper method, passing null context and null connection, will call protected Published Business ServiceMethodName.

Parameters:

`vo` - the value object representing input data for getAddressBook

Returns:

showVO the response data from the query of V0101XPI

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

Class AddressBookManager (6 of 7)

---

getAddressBook

```
protected ShowAddressBook getAddressBook(oracle.e1.bssvfoundation.base.IContext context,
                                         oracle.e1.bssvfoundation.connection.IConnection connection,
                                         GetAddressBook vo)
                                  throws oracle.e1.bssvfoundation.exception.BusinessServiceException
```

Protected method for AddressBookManager Published Business Service. getAddressBook will call make calls to BSSV classes for completing business process.

Parameters:

`vo` - the value object representing input data for getAddressBook.
`context` - conditionally provides the connection for the database operation and logging information
`connection` - can either be an explicit connection or null. If null the default connection is used.

Returns:

confirmVO the response data from the business process getAddressBook.

Throws:

`oracle.e1.bssvfoundation.exception.BusinessServiceException`

Class AddressBookManager (7 of 7)

Viewing Value Object and Field Information

Access the javadoc folder in your JD Edwards EnterpriseOne install directory and double click on the overview-summary.html file.

Click the link for the business service you want to review. For example, to view the value objects for the AddressBookManager business service, click oracle.e1.bssv.JP010000.valueobject.

Your web browser opens a new page that lists each value object that is associated with the business service.

Overview **Package** Class Use Tree Deprecated Index Help

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES All Classes

EnterpriseOne
Service Business Functions

Package oracle.e1.bssv.JP010000.valueobject

Class Summary	
Address	Address is part of the Exposed Input value object
AddressBook	
AddressBookResult	
AlternateAddress	Exposed Input Value Object that represents an alternate address and the action to be peformed for it.
AlternateAddressRecord	Exposed Input Value Object that represents an Alternate Address record
CategoryCodesAddressBook	
CategoryCodesContact	
Classifications	
ConfirmProcessAddressBook	TODO: Java Doc comments for Value Object here
ConfirmProcessContact	ConfirmProcessContact Output VO
Contact	Exposed input value object of Contact information
ContactRecord	Exposed Input Value Object that represents a contact record that will be returned to the consumer
ElectronicAddress	Published Value Object that represents a single electronic address and the action to be performed on it.
ElectronicAddressRecord	Exposed Input Value Object that represents an Electronic Address Record
GetAddress	
GetAddressBook	
GetContact	Exposed Input Value Object that provides input fields used to query the contact table.
Parent	
PhoneNumber	Exposed Input Value Object that represents a single phone number and the action to be peformed for it.
PhoneNumberRecord	Exposed Input Value Object that represents an Phone Number record
ProcessAddressBook	

Package oracle.e1.bssv.JP010000.valueobject page

Click on the value object that you want to view. The web browser displays a summary of the value object, along with a summary of each field that is included in the value object.

Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

EnterpriseOne
Service Business Functions

oracle.e1.bssv.JP010000.valueobject

Class ContactRecord

```
java.lang.Object
  └oracle.e1.bssvfoundation.base.ValueObject
      └oracle.e1.bssv.JP010000.valueobject.GetContact
          └oracle.e1.bssv.JP010000.valueobject.ContactRecord
```

All Implemented Interfaces:

Serializable

```
public class ContactRecord
extends GetContact
implements Serializable
```

Exposed Input Value Object that represents a contact record that will be returned to the consumer

See Also:

Serialized Form

Class ContactRecord page (1 of 7)

Field Summary

private AlternateAddressRecord[]	alternateAddress Alternate Address
private ElectronicAddressRecord[]	electronicAddress Entity
private PhoneNumberRecord[]	phone Phone

Constructor Summary

ContactRecord() Default public constructor for instantiating: ContactRecord

Class ContactRecord page (2 of 7)

Method Summary

AlternateAddressRecord[]	getAlternateAddress()
AlternateAddressRecord	getAlternateAddress(int i)
ElectronicAddressRecord[]	getElectronicAddress()
ElectronicAddressRecord	getElectronicAddress(int i)
PhoneNumberRecord[]	getPhone() class in oracle.e1.bssv.JP010000.valueobject
PhoneNumberRecord	getPhone(int i)
void	setAlternateAddress(AlternateAddressRecord[] alternateAddress)
void	setAlternateAddress(int i, AlternateAddressRecord alternateAddress)
void	setElectronicAddress(ElectronicAddressRecord[] electronicAddress)
void	setElectronicAddress(int i, ElectronicAddressRecord electronicAddress)
void	setPhone(int i, PhoneNumberRecord phone)
void	setPhone(PhoneNumberRecord[] phone)

Class ContactRecord page (3 of 7)

Methods inherited from class oracle.e1.bssv.JP010000.valueobject.GetContact

getCategoryCodesContact, getContactID, getContactTitle, getContacTypeCode, getDayOfBirth, getDescription1, getDisplaySequence, getEntity, getEntityNameContact, getEntityNameSecondary, getFunctionCode, getGenderCode, getGiveName, getMailingNameSecondary, getMiddleName, getMonthOfBirth, getNameMailing, getNickName, getPreferredContactMethod, getPrimaryContactCode, getRemark, getSalutationName, getSurname, getTypeCode, getYearOfBirth, setCategoryCodesContact, setContactID, setContactID, setContactTitle, setContacTypeCode, setDayDateOfBirth, setDayOfBirth, setDescription1, setDisplaySequence, setDisplaySequence, setEntity, setEntityNameContact, setEntityNameSecondary, setFunctionCode, setGenderCode, setGiveName, setMailingNameSecondary, setMiddleName, setMonthDateOfBirth, setMonthOfBirth, setNameMailing, setNickName, setPreferredContactMethod, setPrimaryContactCode, setRemark, setSalutationName, setSurname, setTypeCode, setYearDateOfBirth, setYearOfBirth

Methods inherited from class oracle.e1.bssvfoundation.base.ValueObject

toString, transformBoolean01, transformBooleanYN, transformToBoolean

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Class ContactRecord page (4 of 7)

Field Detail

alternateAddress

```
private AlternateAddressRecord[] alternateAddress
```

Alternate Address

This object contains and array of alternate addresses

---

phone

```
private PhoneNumberRecord[] phone
```

Phone

This object contains and array of Phones

---

electronicAddress

```
private ElectronicAddressRecord[] electronicAddress
```

class in oracle.e1.bssv.JP010000.valueobject

Entity

This object contains and array of entities

Class ContactRecord page (5 of 7)

Constructor Detail

ContactRecord

```
public ContactRecord()
```

Default public constructor for instantiating: ContactRecord

Method Detail

setAlternateAddress

```
public void setAlternateAddress(AlternateAddressRecord[] alternateAddress)
```

---

setAlternateAddress

```
public void setAlternateAddress(int i,
                                AlternateAddressRecord alternateAddress)
```

---

getAlternateAddress

```
public AlternateAddressRecord[] getAlternateAddress()
```

---

getAlternateAddress

```
public AlternateAddressRecord getAlternateAddress(int i)
```

Class ContactRecord page (6 of 7)

setPhone

```
public void setPhone(PhoneNumberRecord[] phone)
```

setPhone

```
public void setPhone(int i,
                     PhoneNumberRecord phone)
```

getPhone

```
public PhoneNumberRecord[] getPhone()
```

getPhone

```
public PhoneNumberRecord getPhone(int i)
```

setElectronicAddress

```
public void setElectronicAddress(ElectronicAddressRecord[] electronicAddress)
```

setElectronicAddress

```
public void setElectronicAddress(int i,
                                 ElectronicAddressRecord electronicAddress)
```

Class ContactRecord page (7 of 7)

To view detailed information about a field or method, click on the field or method name link.

Note. When viewing fields, the field names are in alphabetical order in the Field Summary table. The Field Detail table lists the fields in the order in which they appear in the source code. This preserves the logical groupings that were established by the programmer.

Class AlternateAddressRecord

```
java.lang.Object
  └oracle.e1.bssvfoundation.base.ValueObject
      └oracle.e1.bssv.JP010000.valueobject.AlternateAddressRecord
```

All Implemented Interfaces:
Serializable

```
public class AlternateAddressRecord
extends oracle.e1.bssvfoundation.base.ValueObject
implements Serializable
```

Exposed Input Value Object that represents an Alternate Address record

See Also:
Serialized Form

Field Summary	
private String	addressLine1 Address Line 1
private String	addressLine2 Address Line 2
private String	addressLine3 Address Line 3
private String	addressLine4 Address Line 4
private String	city City
private String	contactAddressTypeCode Type - Address

Class AlternateAddressRecord page - Field Summary table.

Field Detail

dateEffective

```
private Calendar dateEffective
```

Date - Beginning Effective

The date on which an address, item, transaction, or table record becomes active. The meaning of this field differs, depending on the program. For example, the effective date could represent the following:

o When a change of address becomes effective

o When a lease becomes effective

o When a price becomes effective

o When the currency exchange rate becomes effective

o When a tax rate becomes effective

---

contactAddressTypeCode

```
private String contactAddressTypeCode
```

Type - Address

A user defined code (01/AT) that identifies the type of address, such as a home address or an office address.

EnterpriseOne Key Field: true
EnterpriseOne Alias: ATYPE
EnterpriseOne field length: 5

Class AlternateAddressRecord page - Field Detail information.

Viewing the Uses of an Object in Javadoc

Using any of the previous tasks in this section, access the object on which you want to inquire. Click the Use link in the navigation bar on the Javadoc page.

Overview Package Class Use Tree Deprecated Index Help

PREV NEXT FRAMES NO FRAMES All Classes

EnterpriseOne Service Business Functions

Uses of Package oracle.e1.bssv.JP010000.valueobject

Packages that use oracle.e1.bssv.JP010000.valueobject	
oracle.e1.bssv.J0100001.valueobject	
oracle.e1.bssv.J0100002.valueobject	
oracle.e1.bssv.J0100003.valueobject	
oracle.e1.bssv.J0100004.valueobject	
oracle.e1.bssv.J0100005.valueobject	
oracle.e1.bssv.J0100006.valueobject	
oracle.e1.bssv.J0100024.valueobject	
oracle.e1.bssv.JP010000	
oracle.e1.bssv.JP010000.valueobject	
oracle.e1.bssv.JP010020.valueobject	

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100001.valueobject
ProcessAddressBook

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100002.valueobject
GetAddressBook

Uses of Package oracle.e1.bssv.JP010000.valueobject page (1 of 5)

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100003.valueobject
ProcessContact

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100004.valueobject
GetContact Exposed Input Value Object that provides input fields used to query the contact table.

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100005.valueobject
PhoneNumber Exposed Input Value Object that represents a single phone number and the action to be peformed for it.

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100006.valueobject
ElectronicAddress Published Value Object that represents a single electronic address and the action to be performed on it.

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.J0100024.valueobject
AlternateAddress Exposed Input Value Object that represents an alternate address and the action to be peformed for it.

Uses of Package oracle.e1.bssv.JP010000.valueobject page (2 of 5)

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.JP010000
ConfirmProcessAddressBook TODO: Java Doc comments for Value Object here
ConfirmProcessContact ConfirmProcessContact Output VO
GetAddressBook
GetContact Exposed Input Value Object that provides input fields used to query the contact table.
ProcessAddressBook
ProcessContact
ShowAddressBook
ShowContact ShowContact Output VO

Uses of Package oracle.e1.bssv.JP010000.valueobject page (3 of 5)

Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.JP010000.valueobject
Address Address is part of the Exposed Input value object
AddressBook
AddressBookResult
AlternateAddress Exposed Input Value Object that represents an alternate address and the action to be peformed for it.
AlternateAddressRecord Exposed Input Value Object that represents an Alternate Address record
CategoryCodesAddressBook
CategoryCodesContact
Classifications
Contact Exposed input value object of Contact information
ContactRecord Exposed Input Value Object that represents a contact record that will be returned to the consumer
ElectronicAddress Published Value Object that represents a single electronic address and the action to be performed on it.
ElectronicAddressRecord Exposed Input Value Object that represents an Electronic Address Record
GetAddress
GetContact Exposed Input Value Object that provides input fields used to query the contact table.

Uses of Package oracle.e1.bssv.JP010000.valueobject page (4 of 5)

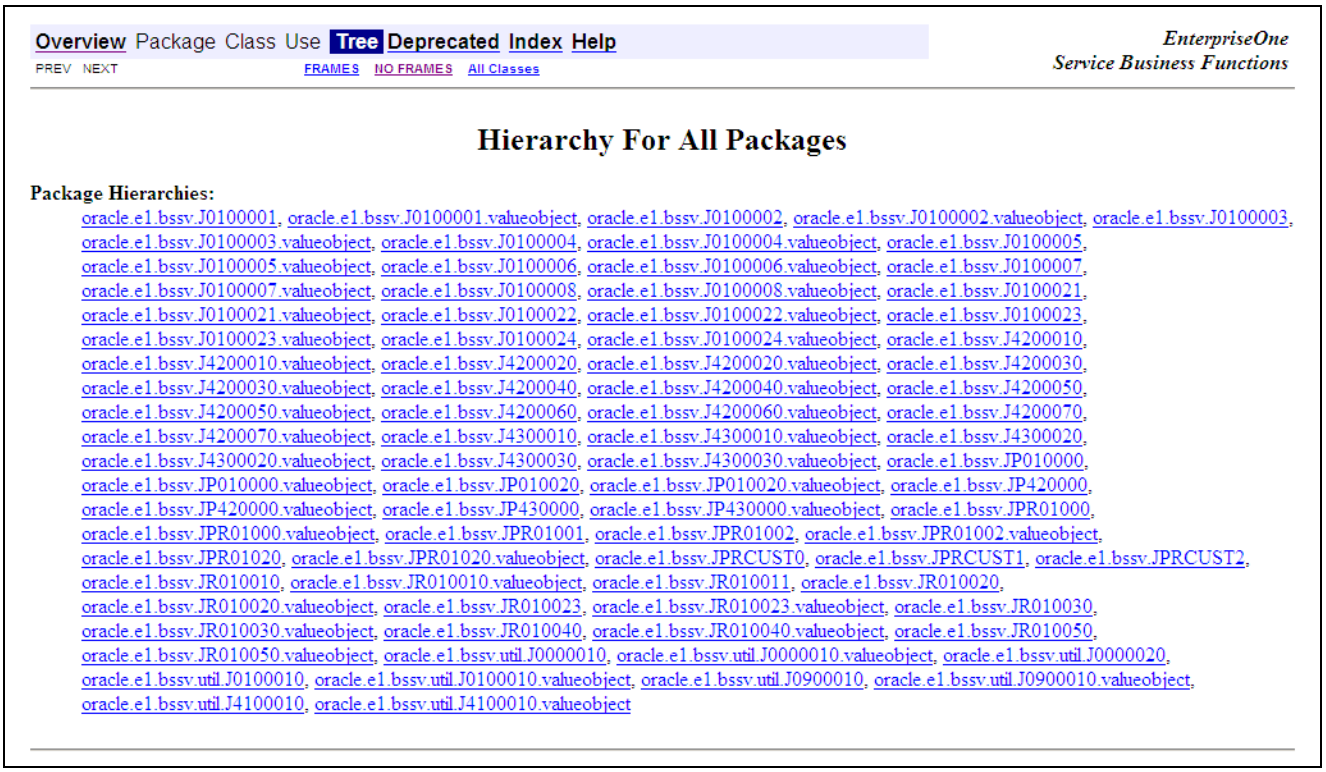
Classes in oracle.e1.bssv.JP010000.valueobject used by oracle.e1.bssv.JP010020.valueobject
ConfirmProcessAddressBook TODO: Java Doc comments for Value Object here
ElectronicAddress Published Value Object that represents a single electronic address and the action to be performed on it.
ElectronicAddressRecord Exposed Input Value Object that represents an Electronic Address Record
PhoneNumber Exposed Input Value Object that represents a single phone number and the action to be peformed for it.
PhoneNumberRecord Exposed Input Value Object that represents an Phone Number record
ProcessAddressBook
Processing

Uses of Package oracle.e1.bssv.JP010000.valueobject page (5 of 5)

Viewing Hierarchy Information

Access the javadoc folder in your JD Edwards EnterpriseOne install directory and double click on the overview-summary.html file. From the main overview page, click the Tree link in the navigation bar. The system displays the hierarchy of all packages.

Note. You can access hierarchy information for a selected package by clicking the Tree link from any package, class or interface page.



Overview Package Class Use **Tree** Deprecated Index Help

PREV NEXT FRAMES NO FRAMES All Classes

EnterpriseOne
Service Business Functions

Hierarchy For All Packages

Package Hierarchies:

oracle.e1.bssv.J0100001, oracle.e1.bssv.J0100001.valueobject, oracle.e1.bssv.J0100002, oracle.e1.bssv.J0100002.valueobject, oracle.e1.bssv.J0100003, oracle.e1.bssv.J0100003.valueobject, oracle.e1.bssv.J0100004, oracle.e1.bssv.J0100004.valueobject, oracle.e1.bssv.J0100005, oracle.e1.bssv.J0100005.valueobject, oracle.e1.bssv.J0100006, oracle.e1.bssv.J0100006.valueobject, oracle.e1.bssv.J0100007, oracle.e1.bssv.J0100007.valueobject, oracle.e1.bssv.J0100008, oracle.e1.bssv.J0100008.valueobject, oracle.e1.bssv.J0100021, oracle.e1.bssv.J0100021.valueobject, oracle.e1.bssv.J0100022, oracle.e1.bssv.J0100022.valueobject, oracle.e1.bssv.J0100023, oracle.e1.bssv.J0100023.valueobject, oracle.e1.bssv.J0100024, oracle.e1.bssv.J0100024.valueobject, oracle.e1.bssv.J4200010, oracle.e1.bssv.J4200010.valueobject, oracle.e1.bssv.J4200020, oracle.e1.bssv.J4200020.valueobject, oracle.e1.bssv.J4200030, oracle.e1.bssv.J4200030.valueobject, oracle.e1.bssv.J4200040, oracle.e1.bssv.J4200040.valueobject, oracle.e1.bssv.J4200050, oracle.e1.bssv.J4200050.valueobject, oracle.e1.bssv.J4200060, oracle.e1.bssv.J4200060.valueobject, oracle.e1.bssv.J4200070, oracle.e1.bssv.J4200070.valueobject, oracle.e1.bssv.J4300010, oracle.e1.bssv.J4300010.valueobject, oracle.e1.bssv.J4300020, oracle.e1.bssv.J4300020.valueobject, oracle.e1.bssv.J4300030, oracle.e1.bssv.J4300030.valueobject, oracle.e1.bssv.JP010000, oracle.e1.bssv.JP010000.valueobject, oracle.e1.bssv.JP010020, oracle.e1.bssv.JP010020.valueobject, oracle.e1.bssv.JP420000, oracle.e1.bssv.JP420000.valueobject, oracle.e1.bssv.JP430000, oracle.e1.bssv.JP430000.valueobject, oracle.e1.bssv.JPR01000, oracle.e1.bssv.JPR01000.valueobject, oracle.e1.bssv.JPR01001, oracle.e1.bssv.JPR01002, oracle.e1.bssv.JPR01002.valueobject, oracle.e1.bssv.JPR01020, oracle.e1.bssv.JPR01020.valueobject, oracle.e1.bssv.JPRCUST0, oracle.e1.bssv.JPRCUST1, oracle.e1.bssv.JPRCUST2, oracle.e1.bssv.JR010010, oracle.e1.bssv.JR010010.valueobject, oracle.e1.bssv.JR010011, oracle.e1.bssv.JR010020, oracle.e1.bssv.JR010020.valueobject, oracle.e1.bssv.JR010023, oracle.e1.bssv.JR010023.valueobject, oracle.e1.bssv.JR010030, oracle.e1.bssv.JR010030.valueobject, oracle.e1.bssv.JR010040, oracle.e1.bssv.JR010040.valueobject, oracle.e1.bssv.JR010050, oracle.e1.bssv.JR010050.valueobject, oracle.e1.bssv.util.J0000010, oracle.e1.bssv.util.J0000010.valueobject, oracle.e1.bssv.util.J0000020, oracle.e1.bssv.util.J0100010, oracle.e1.bssv.util.J0100010.valueobject, oracle.e1.bssv.util.J0900010, oracle.e1.bssv.util.J0900010.valueobject, oracle.e1.bssv.util.J4100010, oracle.e1.bssv.util.J4100010.valueobject

Hierarchy For All Packages page (1 of 2)

Class Hierarchy

- java.lang.**Object**
  - oracle.e1.bssvfoundation.base.BusinessService
    - oracle.e1.bssv.J0100001.**AddressBookProcessor**
    - oracle.e1.bssv.J0100002.**AddressBookQueryProcessor**
    - oracle.e1.bssv.J0100024.**AlternateAddressesProcessor**
    - oracle.e1.bssv.J0100003.**ContactProcessor**
    - oracle.e1.bssv.J0100004.**ContactQueryProcessor**
    - oracle.e1.bssv.J0100023.**CustomerCreditQueryProcessor**
    - oracle.e1.bssv.J4200040.**CustomerItemPriceQueryProcessor**
    - oracle.e1.bssv.J0100021.**CustomerProcessor**
    - oracle.e1.bssv.J0100022.**CustomerQueryProcessor**
    - oracle.e1.bssv.J0100006.**ElectronicAddressesProcessor**
    - oracle.e1.bssv.J0100008.**ElectronicAddressesQueryProcessor**
    - oracle.e1.bssv.util.J0100010.**EntityProcessor**
    - oracle.e1.bssv.util.J0900010.**GLAccountProcessor**
    - oracle.e1.bssv.util.J4100010.**InventoryItemIdProcessor**
    - oracle.e1.bssv.J4200060.**ItemListPriceQueryProcessor**
    - oracle.e1.bssv.J4200030.**ItemPriceAndAvailabilityQueryProcessor**
    - oracle.e1.bssv.util.J0000020.**NetChangeProcessor**
    - oracle.e1.bssv.J0100005.**PhonesProcessor**
    - oracle.e1.bssv.J0100007.**PhonesQueryProcessor**
    - oracle.e1.bssv.util.J0000010.**ProcessingVersionProcessor**
    - oracle.e1.bssvfoundation.base.PublishedBusinessService (implements javax.xml.rpc.server.ServiceLifecycle)
      - oracle.e1.bssv.JP010000.**AddressBookManager**
      - oracle.e1.bssv.JP010020.**CustomerManager**
      - oracle.e1.bssv.JP430000.**ProcurementManager**
      - oracle.e1.bssv.JPR01000.**RI_AddressBookManager**
        - oracle.e1.bssv.JPRCUST0.**RI_CustomABManager**
      - oracle.e1.bssv.JPR01002.**RI_AddressBookStagingManager**
        - oracle.e1.bssv.JPRCUST2.**RI_CustomAddressBookStagingManager**
      - oracle.e1.bssv.JPR01001.**RI_AddressBookTransactionManager**
        - oracle.e1.bssv.JPRCUST1.**RI_CustomAddressBookManager**
      - oracle.e1.bssv.JPR01020.**RI_CustomerManager**
      - oracle.e1.bssv.JP420000.**SalesOrderManager**

Hierarchy For All Packages page (2 of 2)

Hierarchy For Package oracle.e1.bssv.JP010000.valueobject

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.**Object**
  - oracle.e1.bssvfoundation.base.ValueObject (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**Address** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**AddressBook** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**AddressBookResult**
    - oracle.e1.bssv.JP010000.valueobject.**AlternateAddress** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**AlternateAddressRecord** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**CategoryCodesAddressBook** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**CategoryCodesContact** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**Classifications** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**Contact** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**ElectronicAddress** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**ElectronicAddressRecord** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**GetAddress** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**GetAddressBook** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**GetContact** (implements java.io.Serializable)
      - oracle.e1.bssv.JP010000.valueobject.**ContactRecord** (implements java.io.Serializable)
    - oracle.e1.bssvfoundation.base.MessageValueObject (implements java.io.Serializable)
      - oracle.e1.bssv.JP010000.valueobject.**ConfirmProcessAddressBook** (implements java.io.Serializable)
      - oracle.e1.bssv.JP010000.valueobject.**ConfirmProcessContact** (implements java.io.Serializable)
      - oracle.e1.bssv.JP010000.valueobject.**ShowAddressBook**
      - oracle.e1.bssv.JP010000.valueobject.**ShowContact** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**Parent**
    - oracle.e1.bssv.JP010000.valueobject.**PhoneNumber** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**PhoneNumberRecord** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**ProcessAddressBook** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**ProcessContact** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**Processing** (implements java.io.Serializable)
    - oracle.e1.bssv.JP010000.valueobject.**RelatedAddress**

Hierarchy for Package oracle.e1.bssv.JP010000.valueobject page.

Reviewing Input and Response Interfaces

This section provides an overview of input and response interfaces.

Understanding Input and Response Interfaces

You use input and response interface information to determine what data must be passed into and out of the JD Edwards EnterpriseOne system in order to make a web service function properly. An input interface and a response interface table are provided for each web service operation. The tables are listed in an appendix to this reference guide, and appear in the order in which the web services appear in the reference guide.

The input interface includes all of the fields that comprise the message that the JD Edwards EnterpriseOne web service sends to the external system. Some of the fields are required, while others are optional. If a field is required, the data must be included in the input message, or the web service will not send the message successfully.

The response interface includes all of the fields that are returned to the JD Edwards EnterpriseOne user via the web service or web service operation.

Reading the Input Interface Tables

The input interface tables list all of the classes and fields that are contained in the interface. A class is a container for a group of fields and other classes. In the interface tables, classes are highlighted, and the fields that are included in the class appear indented in the rows below the highlighted class name.

The input interface tables also specify which fields are required when performing a specific action. If a field is required, then a *Y* appears in the table under the specified action column for that field. These actions are included in the table:

- Add (Column heading A in the tables)
- Change (Column heading C in the tables)
- Delete or Cancel (Column heading D/C in the tables)
- Inquire (Column heading I in the tables)

For example, if a value is required in the currencyCode field when adding a record, the table lists a *Y* in the currencyCode field row under the column titled A. This table illustrates the example:

Class	Field	Data Type	A	C	D/C	I	Key
Financial							
	currencyCode	String	Y				Y
	amountAdjustment	BigDecimal					

In some circumstances, one or more fields within a particular class are required. Typically, the user can select which of the fields within the category to pass in. For example, many web services require a value for the Entity class, which can be passed in using one or more of these fields:

- entityID
- entityLongId
- entityTaxId

In instances where one of the fields within a class is required, the required information is listed in the row that contains the class heading. This table shows an input interface that requires one of the fields in the Entity - customer class to be passed in:

Class	Field	Data Type	A	C	D/C	I	Key
Entity - customer			Y				Y
	entityID	Integer					
	entityLongId	String					
	entityTaxId	String					

The input interface tables also specify the data type for each field. Valid data types include:

- Integer
- String

- Big Decimal
- Calendar
- Boolean

Lastly, the input interface tables specify which of the fields in the interface are key fields. A key field is a field or set of fields in a database table that together form a single unique identifier for a database record. Depending on the transaction, a key field might not be a required input field, as in the case of a query or if the key is generated by the transaction.

Reading the Response Interface Tables

The response interface tables include the classes and fields that are included in the return message that the web service delivers.

Note. Depending on the data that is available in either the JD Edwards EnterpriseOne or the third-party system, it is possible that some or all of the fields will contain blank values.

Please refer to the documentation for each specific web service operation to determine which fields, if any, will always contain a value if the operation completes successfully.

The response interface tables also specify the data type for each field.

See Also

Appendix A, "Appendix A: Input and Response Interfaces," page 177

CHAPTER 3

Accounts Payable Manager Web Service

This chapter provides an overview of the Accounts Payable Manager web service, lists prerequisites, and discusses these web service operations:

- getVoucher
- processVoucher

Understanding the Accounts Payable Manager Web Service

The Accounts Payable Manager web service (JP040000) manages the processing of accounts payable-related web service operations. This table includes a description of the accounts payable web service operations:

Operation	Description
getVoucher	Use this operation to retrieve and review accounts payable vouchers from the JD Edwards EnterpriseOne Accounts Payable system.
processVoucher	Use this operation to add, delete or void accounts payable vouchers in the JD Edwards EnterpriseOne Accounts Payable system.

Accessing Javadoc for the Accounts Payable Manager Web Service

To access Javadoc for the Accounts Payable Manager web service and its related operations, review these Javadoc packages:

- JP040000 (AccountsPayableManager)
- J0400001 (processVoucher)
- J0400002 (getVoucher)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7.

See Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before you can use any of the accounts payable web service operations, you must install and configure the JD Edwards EnterpriseOne Accounts Payable system.

See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Setting Up the Accounts Payable System".

getVoucher

This section provides an overview of the getVoucher web service operation and lists prerequisites.

Understanding the getVoucher Web Service Operation

The getVoucher web service operation is a database query operation that enables consumers to retrieve accounts payable vouchers from the JD Edwards EnterpriseOne Accounts Payable system.

The operation retrieves records based on the selection criteria that is passed in. If no selection criteria is passed in, the operation attempts to retrieve all records from the EnterpriseOne database, using the business service property to limit the number of records returned. If the operation encounters errors, processing stops and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data. This table includes information about the business service properties that the getVoucher operation uses:

Group	Key	Description	Default Value
J0400002	J0400002_MAX_ROWS	Use this business service property to specify the maximum number of records the operation returns for the query.	*0:* Return All Records

Note. It is recommended that this business service property is configured and used. Also, it is strongly recommended that the consumer pass in selection criteria when performing a query. If these two recommendations are not followed, the operation attempts to fetch all the records from the database, which can significantly impact performance.

Also be aware that this operation performs multiple data retrievals from the EnterpriseOne database. Initially, records are selected from the Accounts Payable Ledger table (F0411) based on the selection criteria that is passed in by the consumer. After the operation retrieves records from the F0411, processing continues, and additional records are retrieved from the Account Ledger table (F0911), based on the data in the F0411 records. This business service property limits the number of records initially retrieved from the F0411. Therefore, the total number of records that are returned to the consumer might be greater than the value that you enter in this property.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getVoucher operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, zero to many records are returned to the consumer based on the selection criteria. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

processVoucher

This section provides an overview of the processVoucher web service operation and lists prerequisites.

Understanding the processVoucher Web Service Operation

The processVoucher web service operation is an inbound transaction operation that enables consumers to add, delete or void vouchers in the JD Edwards EnterpriseOne Accounts Payable system.

If the operation completes successfully, the consumer receives a success message. If the operation encounters errors, processing stops and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes vouchers. This table includes information about the business service properties that the processVoucher operation uses:

Group	Key	Description	Default Value
J0400001	J0400001_GL_VERSION	Use this business service property to specify the version of the P0400074 that the operation uses to process voucher information.	*ZJDE0001*
J0400001	J0400001_ GL_VCHR_ MBF_VERSION	Use this business service property to specify the version of the P0900049 that the operation uses to process voucher information.	*ZJDE0001*

Note. The published value object includes a field in the general ledger array that enables the user to enter a version of the P0900049 program for the F0911FSEditLine function. Though the user can enter a different version for each general ledger record, the operation uses only the first version. The operation first tries to retrieve the processing version from the value object. If no value is entered in the value object, the operation tries to retrieve a version from the business service properties. If no version is found, the operation uses version ZJDE0001.

The header class of the published value object also enables the user to specify a version of theP0400047 program, which is used for the F0411FSBeginDoc and F0411FSEditLine functions. Again, the operation first tries to retrieve the version from the value object. If no version is found, the operation then tries to retrieve a version from the business service properties. If no value is found, the operation uses version ZJDE0001.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the processVoucher operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, a success message is returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany Additionally, if you are adding a voucher, the operation returns two arrays; one for the voucher records added and one for the general ledger records. In addition to the fields listed, the voucher array returns non-zero values for the DocumentPayItem field. It also returns a value for the DocumentPayItemExtension field, but that value can be 0. The general ledger array returns a value in the DocumentLineNumber field.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. If the system encounters errors during processing, no data in the JD Edwards EnterpriseOne system is updated. No manual updates are required.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 4

Address Book Manager Web Service

This chapter provides an overview of the Address Book Manager web service, lists a prerequisite, and discusses these web service operations:

- getAddressBook
- processAddressBook
- getContact
- processContact

Understanding the Address Book Manager Web Service

The AddressBookManager web service (JP010000) manages the processing of address book-related web service operations. This table includes a description of the address book web service operations:

Operation	Description
getAddressBook (J0100002)	Use this operation to retrieve and review address, phone, and electronic mail address information from the JD Edwards EnterpriseOne Address Book system.
processAddressBook (J0100001)	Use this operation to add, change, or delete address book records in the JD Edwards EnterpriseOne Address Book system.
getContact (J0100004)	Use this operation to retrieve and review contact information from the JD Edwards EnterpriseOne Address Book system.
processContact (J0100003)	Use this operation to add, change, or delete contact records in the JD Edwards EnterpriseOne Address Book system.

Accessing Javadoc for the Address Book Manager Web Service

To access Javadoc for the Address Book Manager web service and its related operations, review these Javadoc packages:

- JP010000 (AddressBookManager)
- J0100001 (processAddressBook)
- J0100002 (getAddressBook)
- J0100003 (processContact)

- J0100004 (getContact)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisite

Before you can use the Address Book Manager web service, or any of the related web service operations, you must first install and configure the JD Edwards EnterpriseOne Address Book system.

See *JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide*, "Setting Up the JD Edwards EnterpriseOne Address Book System".

Additionally, you must set up business service properties that are used by all Address Book web service operations that process phone and electronic mail address information. This table lists the business service properties that must be set up:

Group	Key	Description	Default Value
J0100005	J0100005_PHONE_ ERROR_PREFIX	Use this business service property to specify the prefix that the system uses for phone error messages.	Phone Record Sent
J0100006	J0100006_ELECADDR_ ERROR_PREFIX	Use this business service property to specify the prefix that the system uses for electronic address error messages.	Electronic Address Record Sent

getAddressBook

This section provides an overview of the getAddressBook web service operation.

Understanding the getAddressBook Web Service Operation

The getAddressBook web service operation is a database query operation that enables consumers to retrieve and review address, phone, and electronic mail information for specified entities from the JD Edwards EnterpriseOne Address Book system.

The operation returns zero to many records if it completes successfully. If the operation encounters errors while processing address information, processing stops and those errors and warnings are returned to the consumer. If the operation encounters errors while processing phone or electronic mail information, the errors are converted to warnings, processing continues, and the warnings are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getAddressBook operation uses:

Group	Key	Description	Default Value
J0100002	J0100002_MAX_ROWS	Use this operation to specify the maximum number of records that the operation returns.	*100*

Note. It is recommended that users configure this business service property, and use a value other than *0* (zero). Setting this property to *0* enables the operation to return all matching records. Additionally, it is recommended that you pass in selection criteria when processing this operation. If you do not pass in selection criteria, and you set this business property to *0,* you might encounter significant performance issues.

Additionally, this operation uses a joined view of the F0101, F0116, and F0150 tables when retrieving data. The system enforces a one-to-one relationship between records from the F0101 and F0116, and a one-to-many relationship between the F0101 and the F0150 during initial record retrieval. After all records have been retrieved, the operation deletes records with duplicate address book numbers (AN8). This business service property is enforced during the initial record retrieval, before duplicate records are deleted. Therefore, the actual number of records that the operation returns might be less than the maximum number of rows specified.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getAddressBook operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system displays zero to many records, based on the selection criteria that the consumer passes in. If the operation fails, an error is returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityID • entityName • entityTypeCode
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

processAddressBook

This section provides an overview of the processAddressBook web service operation.

Understanding the processAddressBook Web Service Operation

The processAddressBook web service operation is an inbound transaction operation that enables consumers to:

- Add records to the JD Edwards EnterpriseOne Address Book system.
- Delete records from the JD Edwards EnterpriseOne Address Book system.
- Change records in the JD Edwards EnterpriseOne Address Book system.

The operation uses the AddressBookMasterMBF business function (N0100041) to process information. If the operation encounters errors while processing address book information, processing stops and all errors and warnings are returned to the consumer. If the operation encounters errors while processing phone or electronic mail information, the errors are converted into warnings, processing continues, and the warnings are returned to the consumer.

Supported Functionality

This section discusses the functionality that the processAddressBook operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

This operation enables consumers to add, change, or delete address book records. The operation also enables users to add, change, or delete the Who's Who records in the Address Book. These records contain a value of *0* (zero) in the Who's Who Line Number field. The operation also processes phone and electronic mail information for the Who's Who record with line number zero.

This operation does not support related person or alternate address processing for an address book record.

See *JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide*, "Entering Address Book Records," Adding Who's Who Information to Address Book Records.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes address book information. This table includes information about the business service properties that the processAddressBook operation uses:

Group	Key	Description	Default Value
J0100001	J0100001_AB_MBF_ VERSION	Use this business service property to specify the version of the AddressBookMasterMBF (N0100041) the operation uses to process address book information.	*ZJDE0001*

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the processAddressBook operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the user receives a return message that includes the records that were added, changed or updated. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityID • entityName • entityTypeCode
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. If the system encounters errors while processing address book information, no data in the JD Edwards EnterpriseOne system is updated. No manual updates are required. If the operation encounters errors while processing phone or electronic mail information, those errors are converted to warnings and the warnings are returned to the consumer. These warnings do not stop the processAddressBook operation from processing, but the phone or electronic address that causes the error is not processed.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

getContact

This section provides an overview of the getContact web service operation.

Understanding the getContact Web Service Operation

The getContact web service operation is a database query operation that enables consumers to retrieve and review contact, alternate address, phone, and electronic mail information for specified contacts from the JD Edwards EnterpriseOne Address Book system.

The operation returns zero to many records if it completes successfully. If the operation encounters errors while processing contact information, processing stops and those errors and warnings are returned to the consumer. If the operation encounters errors while processing phone or electronic mail information, the errors are converted to warnings, processing continues, and the warnings are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getContact operation uses:

Group	Key	Description	Default Value
J0100004	J0100004_MAX_ROWS	Use this operation to specify the maximum number of records that the operation returns.	*100*

Note. It is recommended that users configure this business service property, and use a value other than *0* (zero). Setting this constant to 0 enables the operation to return all matching records. Additionally, it is recommended that you pass in selection criteria when processing this operation. If you do not pass in selection criteria, and you set this business property to 0, you might encounter significant performance issues.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getAddressBook operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system displays zero to many records, based on the selection criteria that the consumer passes in. If the operation fails, an error is returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityID • contactID • entityNameContact
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

processContact

This section provides an overview of the processContact web service operation.

Understanding the processContact Web Service Operation

The processContact web service operation is an inbound transaction operation that enables consumers to add, change, or delete records to and from these JD Edwards EnterpriseOne tables:

- Who's Who table (F0111)
- Phones table (F0115)
- Electronic Addresses table (F01151)
- Alternate Addresses table (F01161)

The operation uses the WhosWhoMBF (N0100087) to process contact information. If the operation encounters errors while processing contact information, processing stops and the errors are returned to the consumer. If the operation encounters errors while processing phone or electronic mail information, the errors are converted to warnings, processing continues, and the warnings are returned to the consumer.

Supported Functionality

This section discusses the functionality that the processContact operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The operation does not support any country-specific localization functionality. Additionally, the operation does not enable users to add, change or delete a contact's related person information.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes contact data. This table includes information about the business service properties that the processContact operation uses:

Group	Key	Description	Default Value
J0100003	J0100003_WHOSWHO_MBF_VERSION	Use this business service property to specify the version of the Who'sWhoMBF (N0100087) that the operation uses to process contact information.	*ZJDE0001*

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the processContact operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the user receives a return message that includes the records that were changed or updated. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityID • contactID
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. If the system encounters errors while processing contact information, no data in the JD Edwards EnterpriseOne system is updated. No manual updates are required. If the operation encounters errors while processing phone or electronic mail information, those errors are converted to warnings and the warnings are returned to the consumer. These warnings do not stop the processContact operation from processing, but the phone or electronic address that causes the error is not processed.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 5

Capital Asset Manager Web Service

This chapter provides an overview of the Capital Asset Manager web service, lists prerequisites, and discusses these web service operations:

- createCapitalAssetConditionBasedAlert
- getCapitalAssetConditionBasedAlert

Understanding the Capital Asset Manager Web Service

The CapitalAssetManager web service (JP130000) manages the processing of capital asset-related web service operations. This table includes a description of the capital asset web service operations:

Operation	Description
createCapitalAssetConditionBasedAlert	Use this operation to add capital asset condition-based alerts within the JD Edwards EnterpriseOne Plant and Equipment Management system.
getCapitalAssetConditionBasedAlert	Use this operation to query the JD Edwards EnterpriseOne database for existing capital asset condition-based alerts.

Accessing Javadoc for the Capital Asset Manager Web Service Operations

To access Javadoc for the Capital Asset Manager web service and its related operations, review these Javadoc packages:

- JP130000 (CapitalAssetManager)
- J1300001 (createCapitalAssetConditionBasedAlert)
- J1300002 (getCapitalAssetConditionBasedAlert)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before using the Capital Asset Manager web service, or any of the related web service operations, you must install and configure the JD Edwards EnterpriseOne Capital Asset Management product including the JD Edwards EnterpriseOne Plant and Equipment Management and JD Edwards EnterpriseOne Condition-Based Maintenance systems.

See *JD Edwards EnterpriseOne Capital Asset Management 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Capital Asset Management".

See *JD Edwards EnterpriseOne Condition-Based Maintenance 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Condition-Based Maintenance".

createCapitalAssetConditionBasedAlert

This section provides an overview of the createCapitalAssetConditionBasedAlert web service operation and discusses how to set processing options for XPI Condition-Based Alerts (P1301150).

Understanding the createCapitalAssetConditionBasedAlert Web Service Operation

The createCapitalAssetConditionBasedAlert web service operation is an inbound transaction operation that enables consumers to process condition-based alert information within the JD Edwards EnterpriseOne system. The consumer can add capital asset condition-based alerts in the JD Edwards EnterpriseOne Plant and Equipment Management system.

If the operation is successful, the system creates condition-based alerts and commits the records.

If the operation fails, the system returns error messages to the consumer.

The createCapitalAssetConditionBasedAlert web service operation calls the XPIConditionBasedAlert master business function (N1301150) to process the capital asset record. When a condition-based alert is successfully created, a condition-based alert key is returned to the caller through the Alert ID. Exceptions are sent to the caller. The minimum required fields to create a condition-based alert in JD Edwards EnterpriseOne are equipmentNumber, alertLevel, eventDate, eventTime, description, and automatedResponseType.

Supported Functionality

This section discusses the functionality that the createCapitalAssetConditionBasedAlert operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The createCapitalAssetConditionBasedAlert operation supports adding capital asset condition-based alerts in the JD Edwards EnterpriseOne Plant and Equipment Management system.

The createCapitalAssetConditionBasedAlert operation does not support updating or deleting capital asset condition-based alerts.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes condition-based alert information. This table includes information about the business service properties that the createCapitalAssetConditionBasedAlert web service operation uses:

Group	Key	Description	Default Value
J1300001	J1300001_CBA_MBF_VERSION	Use this business service property to specify which version of the XPI Condition-Based Alerts program (P1301150) the operation uses.	*ZJDE0001*
J1300001	J1300001_PREFIX_1	Use this business service property to specify the prefix value the operation uses for error messages.	*Capital Asset Send In*

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

The following table includes information that can help determine whether the createCapitalAssetConditionBasedAlert operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system does not return an error message. If the operation succeeds and returns records, the system returns non-zero values for these fields: • alertId • description • alertLevelCode • alertStatusCode • automatedResponseTypeCode • notificationRecipient
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	No. This operation does not reserve records.

Setting Processing Options for XPI Condition-Based Alerts (P1301150)

Processing options enable you to specify the default processing for programs and reports.

For programs, you can specify options such as the default values for specific transactions, whether fields appear on a form, and the version of the program that you want to run.

Process

1. Equipment Number Format

Specify how the system validates the equipment number. Values are:

1: Use the equipment number. This is the default value.

2: Use the unit number.

3: Use the serial number.

Versions

1. Condition-Based Alerts Revisions (P1311) Version

Specify the version that the system uses for the Condition-Based Alerts Revisions program (P1311). If you leave this processing option blank, the system uses the ZJDE0001 version.

2. Condition-Based Alerts Processor (R1312) Version

Specify which version of the Condition-Based Alerts Processor program (R1312) the system uses when processing the automated responses of condition-based alerts. If you leave this processing option blank, the system does not process the automated responses of condition-based alerts. Note: If you leave this processing option blank, you must run a version of the Condition-Based Alerts Processor program (R1312) in order to complete the automated response-type processing.

getCapitalAssetConditionBasedAlert

This section provides an overview of the getCapitalAssetConditionBasedAlert web service operation.

Understanding the getCapitalAssetConditionBasedAlert Web Service Operation

The getCapitalAssetConditionBasedAlert web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne Plant and Equipment Management system to retrieve existing capital asset data. The getCapitalAssetConditionBasedAlert operation invokes a database operation to retrieve capital asset information from the Condition-Based Alerts table (F1310) in JD Edwards EnterpriseOne based on the selection criteria specified in the value object.

The getCapitalAssetConditionBasedAlert web service operation allows an open query.

Note. An open query of the JD Edwards EnterpriseOne database impacts system performance. It is strongly recommended that you set the Max Rows business service property to a value other than *0* (zero). If you leave this value set to *0,* the system returns all matching records. Additionally, it is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database.

If the operation is successful, the system returns zero to many records to the consumer. You can specify the maximum number of records to return during a query using the Max Rows business service property.

If the operation fails, the system returns an error message to the consumer.

The getCapitalAssetConditionBasedAlert web service operation queries the JD Edwards EnterpriseOne condition-based alert information in a real-time fashion. The web service operation, getCapitalAssetConditionBasedAlert, is called by the source system. This web service operation consumes a published interface, GetCapitalAssetConditionBasedAlert, which exposes the selection criteria for retrieving the condition-based alert information from the F1310 table in JD Edwards EnterpriseOne. The fields in GetCapitalAssetConditionBasedAlert are used to set the selection in the getCapitalAssetConditionBasedAlert integration. The web service operation returns results in the ShowCapitalAssetConditionBasedAlert published interface.

These fields are optional to process the getCapitalAssetConditionBasedAlert integration:

- alert
  - alertId
  - description
  - problemDescription
  - alertLevelCode
  - alertStatusCode
  - dateEstimatedStart
  - assignedWorkOrder
  - serviceTypeCode
  - dateRequested
- equipment
  - itemAsset
  - unitNumber
  - serialNumber
  - measurementLocationCode
  - itemTestId
- site
  - entityId
  - entityLongId
  - entityTaxId
- customer
  - entityId
  - entityLongId
  - entityTaxId
- originator
  - entityId

- entityLongId
- entityTaxId

• notificationRecipient
- entityId
- entityLongId
- entityTaxId

• investigationRecipient
- entityId
- entityLongId
- entityTaxId

Note. The system adjusts the date time stamp entered to Coordinated Universal Time (UTC), which the system determines using the time zone on your business service server, unless an offset is used. For example, if the business service server is GMT-7, entering 2007-01-18T00:00:00.000 results in an adjustment to this value of -7 hours, and the actual value that passes in to JD Edwards EnterpriseOne becomes 2007-01-17T17:00:00.000. An offset can be passed in with the original value so that these adjustments do not occur, such as 2007-01-18T00:00:00.000-7:00, which offsets by 7 hours from UTC, resulting in the date being passed into JD Edwards EnterpriseOne as 2007-01-18T00:00:00.000.

Supported Functionality

This section discusses the functionality that the getCapitalAssetConditionBasedAlert operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getCapitalAssetConditionBasedAlert operation supports a database query operation that enables consumers to query the JD Edwards EnterpriseOne Plant and Equipment Management system to retrieve existing capital asset data.

The getCapitalAssetConditionBasedAlert operation does not support the wildcard (asterisk (*)) for search criteria.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes capital asset information. This table includes information about the business service properties that the getCapitalAssetConditionBasedAlert web service operation uses:

Group	Key	Description	Default Value
J1300002	J1300002_F1310_MAX_GRID_ROWS_RETURNED	Use this business service property to define the maximum number of rows that the operation returns when querying the JD Edwards EnterpriseOne database.	*100*

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0,* the system returns all matching records. Additionally, it is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

The following table includes information that can help determine whether the getCapitalAssetConditionBasedAlert operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system does not return an error message. If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • alertId • description • alertLevelCode • alertStatuscode • automatedResponseTypeCode • dateRequested • timeRequested • eventTimeUTC • itemAsset The operation may complete successfully without returning rows because the selection criteria did not match any records in the database or an open query was performed and there were no records. This is considered successful and provides valid information.

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

CHAPTER 6

Customer Manager Web Service

This chapter provides an overview of the Customer Manager web service, lists prerequisites, and discusses these web service operations:

- getCustomer
- getCustomerCreditInformation
- processCustomer

Understanding the Customer Manager Web Service

The Customer Manager web service (JP010020) manages the processing of customer-related web service operations. This table includes a description of the customer web service operations:

Operation	Description
getCustomer (J0100022)	Use this operation to retrieve and review customer information, including address, phone numbers, electronic addresses, and contact information, from the JD Edwards EnterpriseOne database.
getCustomerCreditInformation (J0100023)	Use this operation to retrieve and review customer credit information from the JD Edwards EnterpriseOne Accounts Receivable system.
processCustomer (J0100021)	Use this operation to add, change, or delete customer information in the JD Edwards EnterpriseOne system.

Accessing Javadoc for the Customer Manager Web Service Operations

To access Javadoc for the Customer Manager web service and its related operations, review these Javadoc packages:

- JP010020 (Customer Manager)
- J0100022 (getCustomer)
- J0100023 (getCustomerCreditInformation)
- J0100021 (processCustomer)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before using this web service, or any of the related operations, you must install and configure the JD Edwards EnterpriseOne Address Book and Accounts Receivable systems. Additionally, you should be familiar with customer and contact information that is stored in the Address Book system.

See *JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide*, "Setting Up the JD Edwards EnterpriseOne Address Book System" and *JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide*, "Entering Address Book Records".

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up the JD Edwards EnterpriseOne Accounts Receivable System" and *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up Customer Master Information".

getCustomer

This section provides an overview of the getCustomer web service operation.

Understanding the getCustomer Web Service Operation

The getCustomer web service operation is a database query operation that enables consumers to retrieve and review customer information from the JD Edwards EnterpriseOne database. This operation retrieves these items for each customer, if the data exists in the database:

- Address book information, including:
  - Phone numbers.
  - Electronic addresses.
- Customer master information.
- Who’s Who line 0 contact information.
- Parent address information.

The operation uses the business view V03012JA to select records. Then the operation uses the AB – Get Parent Address business function (B0100002) to retrieve the parent address for each record the operation returns. If the operation is successful, zero to many records are returned. If the operation encounters errors, processing stops and the errors are returned to the consumer.

Supported Functionality

This section discusses the functionality that the getCustomer operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getCustomer operation does not retrieve related-person information for a specified customer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the database. This table includes information about the business service properties that the getCustomer operation uses:

Group	Key	Description	Default Value
J0100022	J0100022_MAX_ROWS	Use this business service property to specify the maximum number of records the operation can return for a query.	*100*

Note. It is recommended that this business service property is configured and used. Also, it is strongly recommended that the consumer pass in selection criteria when performing a query. If these two recommendations are not followed, the operation attempts to fetch all the records from the database, which can significantly impact performance.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getCustomer operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, zero to many records are returned, based on the selection criteria. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityName • entityTypeCode • company

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

getCustomerCreditInformation

This section provides an overview of the getCustomerCreditInformation web service operation and lists a prerequisite.

Understanding the getCustomerCreditInformation Web Service Operation

The getCustomerCreditInformation web service operation is an inbound transaction operation that enables consumers to retrieve customer credit information from the JD Edwards EnterpriseOne database.

The operation uses several business functions, including the CreditCheckProcessing business function (B4200420,) to process credit information. If the operation is successful, credit information for the specified entity is returned to the consumer. If the operation encounters errors, processing stops and the errors are returned to the consumer.

Implementation Details

This table includes information that can help determine whether the getCustomerCreditInformation operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, records containing credit information for the specified entities are returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityID • creditHoldExempt

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation does not perform transactions. No data in the EnterpriseOne system is updated when this operation processes. If you encounter errors when processing this operation, review the error messages, along with your selection criteria, and retry your query. Additionally, you can verify that records matching your selection criteria exist in the EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you use this operation, you must enter customer credit information.

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up Credit and Collections Management".

processCustomer

This section provides an overview of the processCustomer web service operation.

Understanding the processCustomer Web Service Operation

The processCustomer web service operation is an inbound transaction operation that enables consumers to add, change, or delete customer records from the EnterpriseOne database.

The operation uses the CustomerMasterMBF business function (N0100042) to process customer information. If the operation is successful, the system returns a completion message to the consumer. If the operation encounters errors, processing stops and the errors are returned to the consumer.

Note. This operation calls the processAddressBook web service operation to process customer address information. If the operation encounters errors while processing address information, processing stops and those errors are returned to the consumer.

See Chapter 4, "Address Book Manager Web Service," page 39.

Supported Functionality

This section discusses the functionality that the processCustomer operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes customer and address information. This table includes information about the business service properties that the processCustomer operation uses:

Group	Key	Description	Default Value
J0100021	J0100021_CUS_MBF_VERSION	Use this business service property to specify the version of the CustomerMasterMBF business function (N0100042) that the operation uses when processing customer data.	*ZJDE0001*
J0100021	J0100021_AB_MBF_VERSION	Use this business service property to specify the version of the AddressBookMasterMBF (N0100041) that the operation uses when processing customer address information. You should enter a different version than you use in the business service properties of the processAddressBook business service.	*ZJDE0001*

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the processCustomer operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the consumer receives a successful return message. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • entityID • entityName • entityTypeCode

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. If the system encounters errors during processing, no data in the JD Edwards EnterpriseOne system is updated. No manual updates are required.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 7

Customer Service Manager Web Service

This chapter provides an overview of the Customer Service Manager web service, lists prerequisites, and discusses these web service operations:

- getServiceOrder
- processServiceOrder
- getCommitmentDateTime

Understanding the Customer Service Manager Web Service

The CustomerServiceManager web service (JP170000) manages the processing of customer service-related web service operations. This table includes a description of the customer service web service operations:

Operation	Description
getServiceOrder	Use this operation to query the JD Edwards EnterpriseOne database for existing service orders, along with labor and parts information.
processServiceOrder	Use this operation to complete these tasks within the JD Edwards EnterpriseOne Service Management system: • Add, change or delete service orders. • Add, change or delete related labor lines. • Add, change or delete related parts lines.
getCommitmentDateTime	Use this operation to calculate the commitment date and time for a service order. The system calculates the commitment information using data from the EnterpriseOne service agreement for the specified customer or site.

Note. In the EnterpriseOne system, work orders are stored in the Work Order Master table (F4801). When you create a work order, you must specify the document type that is associated with the work order. The system uses the value in the Document Type field (DCTO) to determine the order type of the work order.

You use the Document Type Maintenance program (P40040) to maintain document type information. Document type records are stored in the Document Type Master table (F40039). When you enter a document type into the system, you can associate that document type with an order type. Order types might include manufacturing orders, equipment orders, and service work orders. To specify that the document type is associated with service work orders, enter *05* in the Order Type field.

The getServiceOrder and processServiceOrder operations process only those work orders that have been defined as *service* work orders. Service work orders must include a document type that is associated with an order type of *05*.

See *JD Edwards EnterpriseOne Service Management 9.0 Implementation Guide*, "Setting Up Service Management".

Accessing Javadoc for the Customer Service Manager Web Service Operations

To access Javadoc for the Customer Service Manager web service and its related operations, review these Javadoc packages:

- JP170000 (CustomerServiceManager)
- J1700010 (processServiceOrder)
- J1700020 (getServiceOrder)
- J1700030 (getCommitmentDateTime)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisite

Before using the Customer Service Manager web service, or any of the related web service operations, you must install and configure the Service Management system. Additionally, you must verify that you are using document types associated with order type *05* to identify service work orders.

See *JD Edwards EnterpriseOne Service Management 9.0 Implementation Guide*, "Setting Up Service Management".

getServiceOrder

This section provides an overview of the getServiceOrder web service operation and lists prerequisites.

Understanding the getServiceOrder Web Service Operation

The getServiceOrder web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne Service Management system to retrieve existing service orders, along with related parts and labor information.

If the operation is successful, the system returns zero to many records to the consumer. You can specify the maximum number of service orders to return during a query using the Max Rows business service property. If the operation fails, the system returns an error message to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns when you query the EnterpriseOne database. This table includes information about the business service properties used by the getServiceOrder operation:

Group	Key	Description	Default Value
J1700020	J1700020_MAX_ROWS	Use this business service property to specify the maximum number of service order records that the operation returns when querying the EnterpriseOne database. **Note.** This business service property limits the number of service order header records that are retrieved. It does not limit the number of labor or parts lines that are retrieved. Therefore, the number of rows that the system returns, including order headers, parts lines and labor lines, could be significantly higher than the value in this constant.	*0:*Return all rows.

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0*, the system returns all matching header records, along with any associated labor and parts records. Additionally, it is recommended that you specify selection criteria when you query the EnterpriseOne database. The system returns an error if you perform a query with no selection criteria and this business service property is set to *0*.

Implementation Details

The following table includes information that can help determine whether the getServiceOrder operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • documentNumber • laborLineNumber • operationTypeCode • businessUnit • partsLineNumber **Note.** If the system does not return any errors or records, verify that you have entered selection criteria and that the Max Rows business service property is set to a value other than *0*.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

processServiceOrder

This section provides an overview of the processServiceOrder web service operation, lists prerequisites, and discusses how to set processing options for CRM Service Order Integration Processing Options (P1702650).

Understanding the processServiceOrder Web Service Operation

The processServiceOrder web service operation is an inbound transaction operation that enables consumers to process service order information within the JD Edwards EnterpriseOne system. The consumer can complete these tasks in the JD Edwards EnterpriseOne Service Management system:

- Add, change or delete service orders.
- Add, change or delete labor lines that are associated with a service order.
- Add, change or delete parts lines that are associated with a service order.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes service order information. This table includes information about the business service properties that the processServiceOrder web service operation uses:

Group	Key	Description	Default Value
J1700010	J1700010_SERVICE_ ORDER_INT	Use this business service property to specify which version of the CRM Service Order Integration Processing Options program (P1702650) the operation uses.	ZJDE0001
J1700010	J1700010_BYPASS_ WARNINGS	Use this business service property to specify whether the system ignores warnings that are generated in the EnterpriseOne system during processing. If you want the operation to stop processing upon receipt of a warning, set this property to *0* or leave it blank. Using this setting, the operation stops processing and returns the warnings to the consumer.	*1:* Bypass warnings.

Implementation Details

The following table includes information that can help determine whether the processServiceOrder operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • laborLineNumber • operationTypeCode • businessUnit • partsLineNumber

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing with auto-commit. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	This operation calls the Work Order Integration business function (N1702650). This function reserves records in the EnterpriseOne system when the operation is changing or deleting existing data.

Setting Processing Options for CRM Service Order Integration Processing Options (P1702650)

You use processing options to specify default values that a program uses when processing information.

Service Order Tab

1. Order Type
Use this processing option to specify the default document type that the system uses when you enter a work order. This code also indicates the origin of the transaction. Enter a value from UDC 00/DT.

2. Service Type
Use this processing option to specify the default order type that the system uses when you enter a work order. The order type indicates the type classification of a work order or engineering change order. You can use work order type as a selection criterion for work order approvals. Enter a value from UDC 00/TY.

3. Priority
Use this processing option to specify the default work order priority that the system uses when you enter a work order. This code indicates the priority of a work order or engineering change order in relation to other orders. Enter a value from UDC 00/PR.

4. Primary Service Provider
Use this processing option to specify the Address Book number of a manager or a planner that the system uses as a default when you enter a work order.

5. Secondary Service Provider
Use this processing option to specify the address book number of the supervisor that the system uses as a default when you enter a work order.

6. Primary Technician
Use this processing option to specify the address book number of a person that the system assigns to do the work when you enter a work order.

7. Secondary Technician
Use this processing option to specify the address book number of an inspector that the system uses as a default when you enter a work order.

8. Type Bill of Material
Use this processing option to specify the default parts list that the system uses when you enter a work order. The parts list is a user defined code that designates the type of bill of material. Enter a value from UDC 40/TB.

9. Type Routing
Use this processing option to specify the labor detail that the system uses as a default when you enter a work order. The system retrieves the labor detail for the work order header and uses it to identify the requested service. Enter a value from UDC 40/TR.

10. Entitlement Checking

Use this processing option to determine whether the system performs entitlement checking and the preferred method. Values are:

Blank: Bypass entitlement checking.

1: Check entitlements using the Entitlement Dates table (F1791).

2: Check entitlements without using the Entitlement Dates table.

11. Category Code 01 through **20. Category Code 10**

Use these processing options to specify user-defined default information that the system uses when creating a work order.

21. Business Unit

Use this processing option to specify the default business unit, which the system uses when it creates a new service order. You must specify a business unit that exists in the Business Unit Master table (F0006).

22. Branch

Use this processing option to specify the default branch that the system uses when it creates a service order. You must specify a branch that exists in the Business Unit Master table (F0006).

23. Default Service Provider

Use this processing option to specify the default values for manager and supervisor addresses on the work order. Values are:

Blank: The system does not supply defaults for the manager and supervisor addresses.

1: The system uses defaults for the manager and supervisor addresses based on this hierarchy:

- First: The system uses the values for the manager and supervisor from processing options.
- Second: The system uses the manager value from the SWM Address Book extension record for the customer.
- Third: The system uses the manager value from the equipment record.
- Fourth: The system uses the values for the manager and supervisor from the Work Order Default Coding File table (F48001), based on category codes 1, 2, and 3 from the work order.

Note. The system uses the Assigned To address value from processing options first, and information from the equipment record second.

getCommitmentDateTime

This section provides an overview of the getCommitmentDateTime web service operation and lists prerequisites.

Understanding the getCommitmentDateTime Web Service Operation

The getCommitmentDateTime web service operation is an inbound transaction operation that enables consumers to calculate the commitment date and time for a given service order. The system calculates the commitment information using the date and time that the specified service order was entered into the system along with the commitment information from the customer or site's service contract, which is stored in the JD Edwards EnterpriseOne Service Management system.

If the system encounters errors during processing, those errors are returned to the consumer.

Supported Functionality

This section discusses the functionality that the getCommittmentDateTime operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

This operation supports time calculations using time zones and daylight savings rules. Thresholds are supported only if the consumer does not pass in a start date.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system calculates commitment information. This table includes information about the business service properties that the getCommitmentDateTime web service operation uses:

Group	Key	Description	Default Value
J1700030	J1700030_UTC_TIMEZON_DAYLIGHTSAVINGSRULE	Use this business service property to specify the UTC time zone and daylight savings rule that the operation uses when displaying the commitment date and time to the consumer. **Note.** You can specify the time zone only, or the time zone and the daylight savings rule. To specify both pieces of information, enter the time zone followed by "|", and then enter the daylight savings rule. For example, enter *26|USA* for time zone *26* and daylight savings rule *USA*.	*26:* Greenwich Mean Time **Note.** Other values for this business service property are stored in UDC table H91/TZ.
J1700030	J1700030_BYPASS_BSFN_WARNINGS	Use this business service property to specify whether the system ignores warnings that are generated in the EnterpriseOne system during processing. If you want the operation to stop processing upon receipt of a warning, set this property to *0* or leave it blank. Using this setting, the operation stops processing and returns the warnings to the consumer.	*1:* Bypass warnings.

Implementation Details

The following table includes information that can help determine whether the getCommitmentDateTime operation is functioning correctly:

Question	Answer
How do I know if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • dateCommitment • timeCommitment

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	No. This operation does not reserve records that are used to calculate commitment information.

Prerequisites

Before using the getCommitmentDateTime operation, service contracts with commitment information must exist in the EnterpriseOne system. Additionally, you must set up service and warranty constants, time zone information, and contact coverage.

See *JD Edwards EnterpriseOne Service Management 9.0 Implementation Guide*, "Managing Contracts" and *JD Edwards EnterpriseOne Service Management 9.0 Implementation Guide*, "Setting Up Service Management," Setting Up Service and Warranty Constants.

CHAPTER 8

Equipment Manager Web Service

This chapter provides an overview of the Equipment Manager web service, lists prerequisites, and discusses these web service operations:

- EquipmentProcessor
- EquipmentQueryProcessor

Understanding the Equipment Manager Web Service

The EquipmentManager web service (JP170001) manages the processing of equipment-related web service operations. This table includes a description of the EquipmentManager web service operations:

Operation	Description
EquipmentProcessor	Use this operation to add new equipment records, or change existing equipment records in the JD Edwards EnterpriseOne system.
EquipmentQueryProcessor	Use this operation to retrieve and review equipment records that are stored in the JD Edwards EnterpriseOne database.

Accessing Javadoc for the EquipmentManager Web Service Operations

To access Javadoc for the EquipmentManager web service and its related operations, review these Javadoc packages:

- JP170001 (EquipmentManager)
- J1700001 (equipmentProcessor)
- J1700002 (equipmentQueryProcessor)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

equipmentProcessor

This section provides an overview of the equipmentProcessor web service operation and lists prerequisites.

Understanding the EquipmentProcessor Web Service Operation

The equipmentProcessor web service operation is an inbound transaction operation that enables consumers to add new equipment records to the JD Edwards EnterpriseOne system, or to change existing equipment records. The operation uses the CRMInstalledBaseProcessing business function (N1702710) to process equipment information.

If the operation completes successfully, all data additions and changes are updated in these JD Edwards EnterpriseOne tables:

- Asset Master table (F1201)
- Equipment Master Extension table (F1217)

If the operation encounters errors while processing, no updates are made to the JD Edwards EnterpriseOne tables, and the error messages are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes equipment data. This table includes information about the business service properties that the EquipmentProcessor operation uses:

Group	Key	Description	Default Value
J1700001	J1700001_EQ_MBF_VERSION	Use this property to specify the version of the Equipment Master Revisions application (P1702) that the CRMInstalledBase business function (N1702710) uses to process equipment information.	ZJDE0001

Implementation Details

This table includes information that can help determine whether the EquipmentProcessor operation is functioning correctly:

Question	Answer

How can I tell if the operation completes successfully?	If the operation completes successfully, all changes or additions are updated in the JD Edwards EnterpriseOne system.
	
	The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.
	
	At a minimum, the system returns non-zero values for these fields:
	
	• actionType
	• assetId
	• description
	• unitNumber
	• serialNumber
	• Customer entity, which includes one or more of these fields:
	  - entityId
	  - entityLongId,
	  - entityTaxId
	• Site entity, which includes one or more of these fields:
	  - entityId
	  - entityLongId,
	  - entityTaxId

If I encounter errors while process a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. If the system encounters errors during processing, no data in the JD Edwards EnterpriseOne system is updated. No manual updates are required.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

EquipmentQueryProcessor

This section provides an overview of the EquipmentQueryProcessor web service operation and lists prerequisites.

Understanding the EquipmentQueryProcessor Web Service Operation

The EquipmentQueryProcessor web service operation is a database query operation that enables consumers to retrieve and review equipment records from the JD Edwards EnterpriseOne system. The operation retrieves data from these EnterpriseOne tables:

- Asset Master table (F1201)
- Equipment Master Extension table (F1217)

If the operation encounters errors, processing stops and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the EquipmentQueryProcessor operation uses:

Group	Key	Description	Default Value
J1700002	J1700002_V1201R_ MAX_GRID_ROWS_ RETURNED	Use this business service property to specify the maximum number of records the operation can return for a query.	*20*

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0*, the system returns all matching records, which could impact performance.

Implementation Details

This table includes information that can help determine whether the EquipmentQueryProcessor operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, zero to many records that match the selection criteria are returned to the consumer. If the operation completes successfully, but finds no matching records, a record count of zero (0) is returned. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • assetId • description • unitNumber • serialNumber • Customer entity, which includes one or more of these fields: - entityId - entityLongId, - entityTaxId • Site entity, which includes one or more of these fields: - entityId - entityLongId, - entityTaxId
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 9

Financial Compliance Manager Web Service

This chapter provides an overview of the Financial Compliance Manager web service, lists prerequisites, and discusses these web service operations:

- getAPProcessingOptions
- getWriteOffProcessingOptions
- getAgingCompanyConstants
- getCustomerCreditLimits
- getGeneralConstants
- getJournalGeneralConstants
- getPolicyEditRules
- getPurchasingToleranceRules
- getSecurityWorkbench

Understanding the Financial Compliance Manager Web Service

The FinancialComplianceManager web service (JP000040) manages the processing of compliance-related web service operations. This table includes a description of the financial compliance web service operations:

Operation	Description
getAPProcessingOptions (J0000041)	Use this operation to retrieve and review the processing option values for these programs in the JD Edwards EnterpriseOne system: • Recycle Recurring Vouchers (R048101) • Purchase Order Receipts (P4312)

Operation	Description
getWriteOffProcessingOptions (J0000042)	Use this operation to retrieve and review the processing options for these JD Edwards EnterpriseOne Accounts Receivable programs: • Standard Receipts Entry (P03B102) • Draft Entry (P03B602) • Speed Receipts Entry (P03B0001) • Invoice Selection Match (R03B50A) • Known Invoice Match With Amount (R03B50D) • Known Invoice Match Without Amount (R03B50E)
getAgingCompanyConstants (J0000043)	Use this operation to retrieve and review company constant records from the Company Constants table (F0010) in the JD Edwards EnterpriseOne system.
getCustomerCreditLimits (J0000044)	Use this operation to retrieve and review accounts receivable information for specified customers, such as credit limits, from the JD Edwards EnterpriseOne system.
getGeneralConstants (J0000045)	Use this operation to retrieve and review constants that determine how the system processes duplicate invoice numbers. This operation retrieves general constants from the General Constants table (F0009) in the JD Edwards EnterpriseOne system.
getJournalGeneralConstants (J0000046)	Use this operation to retrieve and review constants that determine how the system processes journal entry information. This operation retrieves journal general constants from the General Constants table (F0009) in the JD Edwards EnterpriseOne system.
getPolicyEditRules (J0000047)	Use this operation to retrieve and review policy edit rule information from the JD Edwards EnterpriseOne Expense Management system.
getPurchasingToleranceRules (J0000048)	Use this operation to retrieve and review tolerance rules from the JD Edwards EnterpriseOne Procurement system.
getSecurityWorkbench (J0000049)	Use this operation to retrieve and review security workbench records from the JD Edwards EnterpriseOne system.

Accessing Javadoc for the Financial Compliance Manager Web Service Operations

To access Javadoc for the Financial Compliance Manager web service and its related operations, review these Javadoc packages:

- JP000040 (FinancialComplianceManager)
- J0000041 (getAPProcessingOptions)
- J0000042 (getWriteOffProcessingOptions)
- J0000043 (getAgingCompanyConstants)

- J0000044 (getCustomerCreditLimits)
- J0000045 (getGeneralConstants)
- J0000046 (getJournalGeneralConstants)
- J0000047 (getPolicyEditRules)
- J0000048 (getPurchasingToleranceRules)
- J0000049 (getSecurityWorkbench)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

getAPProcessingOptions

This section provides an overview of the getAPProcessingOptions web service operation and lists prerequisites.

Understanding the getAPProcessingOptions Web Service Operation

The getAPProcessingOptions web service operation is a business function query operation that enables consumers to retrieve and review the processing option values for these programs in the JD Edwards EnterpriseOne system:

- Recycle Recurring Vouchers (R048101)
- Purchase Order Receipts (P4312)

You can retrieve processing options for selected versions, or for all versions. If the operation encounters errors while processing, those errors are returned to the consumer.

Implementation Details

This table includes information that can help determine whether the getAPProcessingOptions operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns all processing option values for the specified programs and versions. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and try your query again. If the operation does not return any records, verify that a version matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisites

Before you can use this operation, you should be familiar with recurring accounts payable vouchers and receipt processing.

See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Processing Accounts Payable Vouchers," Processing Recurring Vouchers.

See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Using Receipt Processing".

getWriteOffProcessingOptions

This section provides an overview of the getWriteOffProcessingOptions web service operation and lists prerequisites.

Understanding the getWriteOffProcessingOptions Web Service Operation

The getWriteOffProcessingOptions web service operation is a business function query operation that enables consumers to retrieve and review the processing options for these JD Edwards EnterpriseOne Accounts Receivable programs:

- Standard Receipts Entry (P03B102)
- Draft Entry (P03B602)
- Speed Receipts Entry (P03B0001)
- Invoice Selection Match (R03B50A)
- Known Invoice Match With Amount (R03B50D)

- Known Invoice Match Without Amount (R03B50E)

You can retrieve processing options for selected versions, or for all versions. If the operation encounters errors while processing, those errors are returned to the consumer.

Implementation Details

This table includes information that can help determine whether the getWriteOffProcessingOptions operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns the processing option values for the selected programs and versions. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and try your query again. If the operation does not return any records, verify that a version matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisites

Before you use this operation, you should have an understanding of the Accounts Receivable system, with specific knowledge of drafts and receipts processing.

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up Accounts Receivable Drafts"; *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up Automatic Receipts Processing"; *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Processing Manual Receipts" and *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Processing Accounts Receivable Drafts".

getAgingCompanyConstants

This section provides an overview of the getAgingCompanyConstants web service operation and lists prerequisites.

Understanding the getAgingCompanyConstants Web Service Operation

The getAgingCompanyConstants web service operation is a database query operation that enables consumers to retrieve and review company constant records from the Company Constants table (F0010) in the JD Edwards EnterpriseOne system. The operation returns zero to many records, based on the selection criteria that the consumer passes in, and on the maximum number of rows to return, as defined in the business service properties.

If the consumer does not pass in selection criteria for the fromCompany and toCompany fields, the operation uses these default values when processing the query:

- fromCompany: *00000*
- toCompany: *99999*

This operation returns to the consumer all errors and warnings encountered during processing.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getAgingCompanyConstants operation uses:

Group	Key	Description	Default Value
J0000043	J0000043_QUERY_MAX_ROWS	Use this business service property to specify the maximum number of records the operation returns for a query.	*100*

Note. It is recommended that you configure and use this business service property, and that you send in selection criteria when performing a query. If you leave this property blank, or set it to 0 (zero), the system does not limit the number or records to return. Therefore, sending in a blank query when this property is not set to limit the number of records returned could cause significant performance issues.

Implementation Details

This table includes information that can help determine whether the getAgingCompanyConstants operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system displays company constant records matching the selection criteria. If no records match the selection criteria, the operation returns a response message with an empty array. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.

Question	Answer
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisites

Before using this operation, you must define accounts receivable company constants information in the JD Edwards EnterpriseOne system.

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up the JD Edwards EnterpriseOne Accounts Receivable System," Setting Up Constants for Accounts Receivable.

getCustomerCreditLimits

This section provides an overview of the getCustomerCreditLimits web service operation and lists prerequisites.

Understanding the getCustomerCreditLimits Web Service Operation

The getCustomerCreditLimits web service operation is a database query operation that enables consumers to retrieve and review accounts receivable information, such as credit limits, for selected customers in the JD Edwards EnterpriseOne system. The operation retrieves information from the Customer Master by Line of Business table (F03012).

If the operation encounters errors, processing stops and all errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system getCustomerCreditLimits operation retrieves data. This table includes information about the business service properties that the getCustomerCreditLimits operation uses:

Group	Key	Description	Default Value
J0000044	J0000044_MAX_ROWS	Use this business service property to specify the maximum number of records that the operation can return for a query.	*100*

Note. It is recommended that you configure and use this business service property, and that you send in selection criteria when performing a query. If you leave this property blank, or set it to 0 (zero), the system does not limit the number or records to return. Therefore, sending in a blank query when this property is not set to limit the number of records returned could cause significant performance issues.

Implementation Details

This table includes information that can help determine whether the getCustomerCreditLimits operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system displays accounts receivable records for the specified customers. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisites

Before you use this operation, verify that credit and collection information for your customers exists in the JD Edwards EnterpriseOne Accounts Receivable system.

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up Customer Master Information" and *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up Credit and Collections Management".

getGeneralConstants

This section provides an overview of the getGeneralConstants web service operation and lists prerequisites.

Understanding the getGeneralConstants Web Service Operation

The getGeneralConstants web service operation is a database query operation that enables consumers to retrieve and review general constants from the General Constants table (F0009) in the JD Edwards EnterpriseOne system. The constant that this operation displays determines how the system handles duplicate invoice numbers.

If the operation encounters errors, processing stops and the errors are returned to the consumer.

Note. The F0009 contains only one record. Therefore, this operation will not return multiple records.

Implementation Details

This table includes information that can help determine whether the getGeneralConstants operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the record that exists in the F0009 table is returned to the consumer. This record includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns a non-zero value for the duplicateInvoiceNumberEdit field.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and try your query again. If the operation does not return any records, verify that a record matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you use this operation, you must set up accounts payable and receivable constants in the JD Edwards EnterpriseOne system.

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up the JD Edwards EnterpriseOne Accounts Receivable System," Setting Up Constants for Accounts Receivable.

See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Setting Up the Accounts Payable System," Setting Up Constants for Accounts Payable.

getJournalGeneralConstants

This section provides an overview of the getJournalGeneralConstants web service operation and lists prerequisites.

Understanding the getJournalGeneralConstants Web Service Operation

The getJournalGeneralConstants web service operation is a database query operation that enables consumers to retrieve and review general constants from the General Constants table (F0009) in the JD Edwards EnterpriseOne system. This operation returns constants that determine how the system processes journal entry information. For example, you can use this operation to review whether batch control or management approval is required, whether postings or invalid accounts are allowed, and whether the system processes intercompany settlements or multicurrency transactions.

If the operation encounters errors, processing stops and the errors are returned to the consumer.

Note. The F0009 contains only one record. Therefore, this operation will not return multiple records.

Implementation Details

This table includes information that can help determine whether the getJournalGeneralConstants operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the record that exists in the F0009 table is returned to the consumer. This record includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • isBatchControlRequired • isBatchManagementApprovalRequired • arePostingsAllowed • areInvalidAccountsAllowed • intercompanySettlements • isMultiCurrencyIntercompanyTransAllowed
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and try your query again. If the operation does not return any records, verify that a record matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you use this operation, you must set up accounts payable and receivable constants in the JD Edwards EnterpriseOne system.

See *JD Edwards EnterpriseOne Accounts Receivable 9.0 Implementation Guide*, "Setting Up the JD Edwards EnterpriseOne Accounts Receivable System," Setting Up Constants for Accounts Receivable.

See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Setting Up the Accounts Payable System," Setting Up Constants for Accounts Payable.

getPolicyEditRules

This section provides an overview of the getPolicyEditRules web service operation and lists prerequisites.

Understanding the getPolicyEditRules Web Service Operation

The getPolicyEditRules web service operation is a database query operation that enables consumers to retrieve and review policy edit rule information from the JD Edwards EnterpriseOne Expense Management system. The operation retrieves information from the Policy Edit Rules table (F09E108).

If the operation encounters errors, processing stops and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getPolicyEditRules operation uses:

Group	Key	Description	Default Value
J0000047	J0000047_MAX_ROWS	Use this operation to specify the maximum number of records that the system returns for a query.	*100*

Implementation Details

This table includes information that can help determine whether the getPolicyEditRulesoperation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, policy edit rule records are returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.

Question	Answer
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service properties and try your query again. If the operation does not return any records, verify that a record matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you use this operation, you must define policy rules in the JD Edwards EnterpriseOne Expense Management system.

See *JD Edwards EnterpriseOne Expense Management 9.0 Implementation Guide*, "Setting Up the Expense Management System," Setting Up Expense Management Policies.

getPurchasingToleranceRules

This section provides an overview of the getPurchasingToleranceRules web service operation and lists prerequisites.

Understanding the getPurchasingToleranceRules Web Service Operation

The getPurchasingToleranceRules web service operation is a database query operation that enables consumers to retrieve and review tolerance rules from the JD Edwards EnterpriseOne Procurement system. The operation retrieves data from the Purchasing Tolerance Rules table (F4322).

If the operation encounters errors, processing stops and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getPurchasingToleranceRules operation uses:

Group	Key	Description	Default Value
J0000048	J0000048_MAX_ROWS	Use this operation to specify the maximum number of records that the system returns for a query.	*100*

Implementation Details

This table includes information that can help determine whether the getPurchasingToleranceRules operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the processing completes successfully, the operation returns zero to many records, based on the query. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service properties and try your query again. If the operation does not return any records, verify that a record matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you use this operation, you must set up tolerance rules in the JD Edwards EnterpriseOne Procurement system.

See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Setting Up the Procurement System," Creating Tolerance Rules.

getSecurityWorkbench

This section provides an overview of the getSecurityWorkbench web service operation and lists prerequisites.

Understanding the getSecurityWorkbench Web Service Operation

The getSecurityWorkbench web service operation is a database query operation that enables consumers to retrieve and review security workbench records from the JD Edwards EnterpriseOne system. The operation retrieves data from the Security Workbench table (F00950).

The operation returns security records for these JD Edwards EnterpriseOne programs:

- Manual Payments (P0413M)
- Work With Payment Groups (P04571)
- Create Payment Groups (R04570)
- Speed Status Change (P0411S)
- Multi Company Voucher (P041016)
- Multiple Voucher (P041017)
- Standard Voucher (P0411)

- Speed Voucher (P0411SV)
- Supplier Master (P04012)

If the operation encounters errors, processing stops and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves data from the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getSecurityWorkbenchoperation uses:

Group	Key	Description	Default Value
J0000049	J0000049_MAX_ROWS	Use this operation to specify the maximum number of records that the system returns for a query.	*0:* Return All Rows

Note. The maximum number of rows that this operation can return is minimal. Leaving this option set to zero does not cause performance issues.

Implementation Details

This table includes information that can help determine whether the getSecurityWorkbenchoperation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the processing completes successfully, the operation returns zero to many records, based on the query. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service properties and try your query again. If the operation does not return any records, verify that a record matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you use this operation, you must set up the security workbench in the JD Edwards EnterpriseOne system.

See *JD Edwards EnterpriseOne Tools 8.98 Security Administration Guide*

CHAPTER 10

Financials Manager Web Service

This chapter provides an overview of the Financials Manager web service, lists prerequisites, and discusses these web service operations:

- getGLAccountNumber
- insertBatchJournalEntry

Understanding the Financials Manager Web Service

The Financials web service (JP090000) manages the processing of financial-related web service operations. This table includes a description of the financial web service operations:

Operation	Description
getGLAccount (J0900001)	Use this operation to retrieve account master information from the JD Edwards EnterpriseOne General Accounting system.
insertBatchJournalEntry (J0900002)	Use this operation to insert a batch of journal entries into the JD Edwards EnterpriseOne General Accounting system.

Accessing Javadoc for the Financials Manager Web Service Operations

To access Javadoc for the Financials Manager web service and its related operations, review these Javadoc packages:

- JP090000 (Financials Manager)
- J0900001 (getGLAccount)
- J0900002 (insertBatchJournalEntry)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before you can use the Financials Manager web service or any of its related operations, you must install and set up the JD Edwards EnterpriseOne General Accounting system.

See *JD Edwards EnterpriseOne General Accounting 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne General Accounting".

getGLAccount

This section provides an overview of the getGLAccount web service operation and lists prerequisites.

Understanding the getGLAccount Web Service Operation

The getGLAccount web service operation is a database query operation that enables consumers to retrieve and review general ledger account records from the Account Master table (F0901) in the JD Edwards EnterpriseOne General Accounting system. If the operation encounters errors during processing, the operation does not complete and the errors are returned to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the operation retrieves records from the Account Master table (F0901). This table includes information about the business service properties that the getGLAccount operation uses:

Group	Key	Description	Default Value
J0900001	J0900001_MAX_ROWS	Use this business service property to specify the maximum number of records that the operation can return to the consumer.	*1000*

Note. It is strongly recommended that consumers set this business service property to a value other than *0* and that search criteria is specified when sending a query. Entering *0* in this property will allow the operation to retrieve all records that match the search criteria. Though the operation allows a query without search criteria, sending in such a query without limiting the number of rows that the operation returns could result in significant performance issues.

Implementation Details

This table includes information that can help determine whether the getGLAccount operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system displays general ledger account records that match the search criteria. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for the accountId field.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This operation does not update data in the JD Edwards EnterpriseOne system. If you encounter errors during processing, review your search criteria and business service property settings and retry your query. No manual updates are necessary if errors are encountered.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

insertBatchJournalEntry

This section provides an overview of the insertBatchJournalEntry web service operation and lists prerequisites.

Understanding the insertBatchJournalEntry Web Service Operation

The insertBatchJournalEntry web service operation is a database insert operation that enables consumers to add journal entry records to the Journal Entry Transactions – Batch table (F0911Z1). The F0911Z1 is a temporary table in the JD Edwards EnterpriseOne General Accounting system. After records have been added to this table, consumers can use JD Edwards EnterpriseOne programs to update and process the journal entries that have been added to the temporary table.

If the operation encounters any errors while attempting to insert records into the F0911Z1 table, those errors are returned to the consumer. Additionally, the F0911Z1 table is restored to its original state and no records are inserted into the table.

Supported Functionality

This section discusses the functionality that the insertBatchJournalEntry operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The insertBatchJournalEntry operation supports only the insertion of journal entry records into the F0911Z1 table. The operation does not support changing, deleting, or processing those records. Consumers must use JD Edwards EnterpriseOne programs to change, delete, or process records that this operation inserts into the table.

See *JD Edwards EnterpriseOne General Accounting 9.0 Implementation Guide*, "Processing Batch Journal Entries".

Implementation Details

This table includes information that can help determine whether the insertBatchJournalEntry operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message the includes the number of rows that were inserted into the F0911Z1 table.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. If the system encounters errors during processing, the F0911Z1 table is returned to its original state and all errors are returned to the consumer. No manual updates are necessary.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 11

Fixed Asset Manager Web Service

This chapter provides an overview of the FixedAssetManager web service, lists prerequisites, and discusses the getFixedAsset web service operation.

Understanding the Fixed Asset Manager Web Service

The FixedAssetManager web service (JP120000) manages the processing of the getFixedAsset web service operation (J1200001). You use this operation to query the JD Edwards EnterpriseOne Fixed Asset system, and review fixed asset records. If the system encounters errors during processing, the Fixed Asset Manager returns those errors to the consumer.

Accessing Javadoc for the Fixed Asset Manager Web Service Operations

To access Javadoc for the Fixed Asset Manager web service and its related operations, review these Javadoc packages:

- JP120000 (Fixed Asset Manager)
- J1200001 (getFixedAsset)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before you use the Fixed Asset Manager web service or the getFixedAsset operation, you must install and configure the JD Edwards EnterpriseOne Fixed Asset system. Additionally, fixed asset records must exist in the EnterpriseOne database.

See *JD Edwards EnterpriseOne Fixed Assets 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Fixed Assets".

getFixedAsset

This section provides an overview of the getFixedAsset web service operation.

Understanding the getFixedAsset Web Service Operation

The getFixedAsset web service operation is a database query operation that enables consumers to retrieve and review fixed asset records that exist in the Asset Master table (F1201) in the JD Edwards EnterpriseOne Fixed Asset system.

If the operation completes successfully, fixed asset records, along with any warnings encountered, are returned to the consumer. If the operation encounters errors during processing, the operation stops processing and returns the errors to the consumer.

Note. If no records, warnings, or errors are returned to the consumer, it is likely that the operation completed successfully, but no records matching the selection criteria exist in the JD Edwards EnterpriseOne database.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system getFixedAsset operation processes and returns data. This table includes information about the business service properties that the getFixedAsset operation uses:

Group	Key	Description	Default Value
J1200001	J1200001_MAX_ROWS	Use this business service property to specify the maximum number of rows that the operation returns to the consumer.	*100*

Note. It is recommended that you set this property to a value other than *0* (zero). Setting this property to *0* enables the operation to retrieve all records that match the selection criteria. Additionally, it is recommended that you specify selection criteria when performing a query. Sending a blank query and setting this option to *0* could significantly impact system performance.

Implementation Details

This table includes information that can help determine whether the getFixedAsset operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns fixed asset records to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns a non-zero value for the assetID field.
If I encounter errors while process a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and your business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 12

Foundation Environment Web Service

This chapter provides an overview of the Foundation Environment web service, lists prerequisites, and discusses the getUserDefinedCode web service operation (J0000030).

Understanding the Foundation Environment Web Service

The Foundation Environment web service (JP000000) manages the processing of the getUserDefinedCode web service operation. You use the getUserDefinedCode web service operation to retrieve and review user defined codes from the JD Edwards EnterpriseOne system.

Accessing Javadoc for the Foundation Environment Web Service Operations

To access Javadoc for the Foundation Environment web service and its related operations, review these Javadoc packages:

- JP000000 (Foundation Environment)
- J0000030 (getUserDefinedCode)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before using this web service, you should be familiar with how user defined codes (UDCs) are used in the JD Edwards EnterpriseOne system.

getUserDefinedCode

This section provides an overview of the getUserDefinedCode web service operation and lists prerequisites.

Understanding the getUserDefinedCode Web Service Operation

The getUserDefinedCode web service operation is a database query operation that enables consumers to retrieve and review user-defined code (UDC) tables that are stored in the JD Edwards EnterpriseOne system.

UDC tables are used throughout the EnterpriseOne system. They enable consumers to store, track, calculate, and process information that is specific to their business operations. The getUserDefinedCode operation enables consumers to search on UDC tables and review information about the tables and the values that are stored in those tables.

Note. When you enter a query, you must specify selection criteria. The operation does not allow a query without selection criteria.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns for a query. This table includes information about the business service property that the getUserDefinedCode operation uses:

Group	Key	Description	Default Value
J0000030	J0000030_V0004A_ MAX_GRID_ROWS_ RETURNED	Use this business service property to specify the maximum number of records that the system returns for a query.	*100*

Note. It is strongly recommended that consumers set this business service property to a value other than *0*. Entering *0* in this property will allow the operation to retrieve all records that match the search criteria, and could cause performance issues.

Implementation Details

This table includes information that can help determine whether the getUserDefinedCode operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns UDC records to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • productCode • userDefinedCodeTable • description1
If I encounter errors while process a transaction, do I need to reverse the transaction?	This operation does not process transactions, therefore, no updates are necessary if you encounter errors. If you receive errors while processing this operation, review your selection criteria and business service properties, and then retry your query. If you do not receive any matching records, review the data in the EnterpriseOne system to verify that the records for which you are searching exist in the database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 13

Inventory Manager Web Service

This chapter provides an overview of the Inventory Manager web service, lists a prerequisite, and discusses these web service operations:

- processSupplierCatalogPrice
- processInventoryItem
- getBranchPlantItem
- getSupplierCatalogPrice
- getItemAvailability
- insertInventoryItemStaging

Understanding the Inventory Manager Web Service

The InventoryManager web service (JP410000) manages the processing of inventory-related web service operations. This table includes a description of the inventory web service operations:

Operation	Description
processSupplierCatalogPrice (J4100002)	Use this operation to add and change supplier catalog prices within the JD Edwards EnterpriseOne Inventory Management system.
processInventoryItem (J4100003)	Use this operation to add and change inventory item master records within the JD Edwards EnterpriseOne Inventory Management system.
getBranchPlantItem (J4100004)	Use this operation to query the JD Edwards EnterpriseOne database for existing item branch/plant records.
getSupplierCatalogPrice (J4100005)	Use this operation to query the JD Edwards EnterpriseOne database for existing supplier catalog prices.
getItemAvailability (J4100001)	Use this operation to query the JD Edwards EnterpriseOne database for existing item availability.
insertInventoryItemStaging (J4100006)	Use this operation to insert inventory item staging records within the JD Edwards EnterpriseOne database.

Accessing Javadoc for the Inventory Manager Web Service Operations

To access Javadoc for the Inventory Manager web service and its related operations, review these Javadoc packages:

- JP410000 (InventoryManager)
- J4100002 (processSupplierCatalogPrice)
- J4100003 (processInventoryItem)
- J4100004 (getBranchPlantItem)
- J4100005 (getSupplierCatalogPrice)
- J4100001 (getItemAvailability)
- J4100006 (insertInventoryItemStaging)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisite

Before using the Inventory Manager web service, or any of the related web service operations, you must install and configure the JD Edwards EnterpriseOne Inventory Management system.

See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Inventory Management".

processSupplierCatalogPrice

This section provides an overview of the processSupplierCatalogPrice web service operation.

Understanding the processSupplierCatalogPrice Web Service Operation

The processSupplierCatalogPrice web service operation is an inbound transaction operation that enables consumers to process supplier catalog price information within the JD Edwards EnterpriseOne system. The consumer can add and change supplier catalog price records in the JD Edwards EnterpriseOne Inventory Management system. The system updates the Supplier Price/Catalog File table (F41061).

If the operation is successful, the system returns a confirmation message to the consumer which includes supplier catalog data.

If the operation fails, the system returns an error message to the consumer. When an error is encountered during processing of the processSupplierCatalogPrice web service operation, any changes to the F41061 that occurred as a result of the web service operation being run are rolled back.

The processSupplierCatalogPrice web service operation verifies that values are provided for both the start and end effective dates. If no value is provided for the start date then today's date is assigned to the start date. If no value is provided for the end date then December 31st of the current year is assigned to the end date. The processSupplierCatalogPrice web service operation retrieves the currency code from the specified supplier or, if the supplier does not exist, the specified business unit's company. If a currency code is specified, the currency code lookup is bypassed. The processSupplierCatalogPrice web service operation retrieves the cost level for a specified item and the appropriate unit of measure (purchasing or primary) from the Item Master table (F4101) as specified in the distribution constants. If a unit of measure is specified, the unit of measure lookup is bypassed.

The table illustrates the supported action codes for the processSupplierCatalogPrice web service operation:

Action Type	Action Code
Add	1, A, I
Update	2, U, C

Supported Functionality

This section discusses the functionality that the processSupplierCatalogPrice operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The operation supports adding new supplier catalog prices and updating existing supplier catalog prices.

This web service operation does not support deleting supplier catalog prices. The operation also does not support adding or modifying supplier catalog prices for items with a cost level of *3*.

Implementation Details

The following table includes information that can help determine whether the processSupplierCatalogPrice operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • supplier • item • currencyCode • unitOfMeasureCode • dateEffectiveStart • dateEffectiveEnd The web service operation always returns an exception in the response if the operation fails to complete successfully.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. If the supplier catalog price action encounters errors, the transaction stops processing and the system rolls back the information in the tables. No manual update is necessary.
Does this operation use record reservation?	No. This operation does not reserve records.

processInventoryItem

This section provides an overview of the processInventoryItem web service operation and lists prerequisites.

Understanding the processInventoryItem Web Service Operation

The processInventoryItem web service operation is an inbound transaction operation that enables consumers to process inventory item information within the JD Edwards EnterpriseOne system. The consumer can add and change inventory item master records in the JD Edwards EnterpriseOne Inventory Management system.

Note. The data provided for item master insertion must be in the JD Edwards EnterpriseOne format as no formatting is performed before item master insertion in JD Edwards EnterpriseOne begins. Also, codes must be values in the JD Edwards EnterpriseOne system.

If the operation is successful, the system returns a confirmation message to the consumer. The messages includes inventory item data.

If the operation fails, the system returns an error message to the consumer.

Supported Functionality

This section discusses the functionality that the processInventoryItem operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The consumer can add and change inventory item master records in the JD Edwards EnterpriseOne Inventory Management system.

The processInventoryItem web service operation does not support the cancellation of inventory items.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes inventory item information. This table includes information about the business service properties that the processInventoryItem web service operation uses:

Group	Key	Description	Default Value
J4100003	J4100003_ITEM_MBF_VERSION	Use this business service property to specify which version of the Item Master program (P4101) the operation uses. See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Entering Item Information," Setting Processing Options for Item Master (P4101).	*ZJDE0001*
J4100003	J4100003_ITEM_STOCKING_TYPE_CODE	Use this business service property to specify the stocking type code that the operation uses for an add request if the code is not provided.	*S*

Implementation Details

The following table includes information that can help determine whether the processInventoryItem operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values in the itemId field.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	This operation reserves the inventory record immediately when an update is requested by the consumer.

Prerequisites

Before using the processInventoryItem operation, you must set the processing options for the Item Master program (P4101).

See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Entering Item Information," Setting Processing Options for Item Master (P4101).

getBranchPlantItem

This section provides an overview of the getBranchPlantItem web service operation.

Understanding the getBranchPlantItem Web Service Operation

The getBranchPlantItem web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne Inventory Management system to retrieve existing branch/plant item records. The operation invokes a database operation to retrieve branch/plant item information from the F4101 and Item Branch File (F4102) tables in JD Edwards EnterpriseOne based on the selection criteria specified in the value object.

Note. The data provided for selection must be in the JD Edwards EnterpriseOne format. No formatting is performed before the query is made. Also, codes must be values in the JD Edward EnterpriseOne system.

If the operation is successful, the system returns zero to many records to the consumer. You can specify the maximum number of records to return during a query using the Max Rows business service property. If the operation fails, the system returns an error message to the consumer.

The getBranchPlantItem web service operation enables source systems to query JD Edwards EnterpriseOne branch/plant item information in a real-time fashion. The query contains fields that can be used to filter the branch/plant item information retrieved by JD Edwards EnterpriseOne. ShowBranchPlant is returned during a successful query.

Supported Functionality

This section discusses the functionality that the getBranchPlantItem operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getBranchPlantItem web service operation enables consumers to query the JD Edwards EnterpriseOne Inventory Management system to retrieve existing branch/plant item records.

The getBranchPlantItem operation does not support the wildcard (asterisk (*)) for search criteria.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes inventory item information. This table includes information about the business service properties that the getBranchPlantItem web service operation uses:

Group	Key	Description	Default Value
J4100004	J4100004_V4102XPI2_MAX_GRID_ROWS_RETURNED	Use this business service property to define the maximum number of rows that the operation returns when querying the JD Edwards EnterpriseOne database.	*100*

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0,* the system returns all matching records. Additionally, it is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database. The system returns an error if you perform a query with no selection criteria and this business service property is set to *0.*

Implementation Details

The following table includes information that can help determine whether the getBranchPlantItem operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • branchPlant • supplier • buyer • itemId
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

getSupplierCatalogPrice

This section provides an overview of the getSupplierCatalogPrice web service operation.

Understanding the getSupplierCatalogPrice Web Service Operation

The getSupplierCatalogPrice web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne Inventory Management system to retrieve existing supplier catalog price information. The operation retrieves supplier catalog price information from fields in the Supplier Price/Catalog File table (F41061).

If the operation is successful, the system returns zero to many records to the consumer. You can specify the maximum number of records to return during a query using the Max Grid Rows Returned business service property. If the operation fails, the system returns an error message to the consumer.

Note. The data provided for selection must be in the JD Edwards EnterpriseOne format. No formatting is performed before the query is made.

Supported Functionality

This section discusses the functionality that the getSupplierCatalogPrice operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getSupplierCatalogPrice web service operation enables consumers to query the JD Edwards EnterpriseOne Inventory Management system to retrieve existing supplier catalog price information.

The getSupplierCatalogPrice operation does not support the wildcard (asterisk (*)) for search criteria.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes supplier catalog price information. This table includes information about the business service properties that the getSupplierCatalogPrice web service operation uses:

Group	Key	Description	Default Value
J4100005	J4100005_F41061_ MAX_GRID_ROWS_ RETURNED	Use this business service property to define the maximum number of rows that the operation returns when querying the JD Edward EnterpriseOne database.	*100*

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0,* the system returns all matching records. Additionally, it is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database.

Implementation Details

The following table includes information that can help determine whether the getSupplierCatalogPrice operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • entityIdSupplier • itemId • itemProduct • itemCatalog • currencyCode • unitOfMeasureCode The operation may complete successfully without returning rows because the selection criteria did not match any records in the database or an open query was performed and there were no records. This is considered successful and provides valid information.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

getItemAvailability

This section provides an overview of the getItemAvailability web service operation.

Understanding the getItemAvailability Web Service Operation

The getItemAvailability web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne Inventory Management system to retrieve item availability information.

The getItemAvailability web service operation calls the InvRealTimeItemAvailability business function (B4101640) to fetch item availability. You must specify itemId, businessUnit, and unitOfMeasure as inputs to the query. If the operation is successful, the system returns item availability data. If the operation fails, the system returns an error message to the consumer.

Supported Functionality

This section discusses the functionality that the getItemAvailability operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getItemAvailability web service operation enables consumers to query the JD Edwards EnterpriseOne Inventory Management system to retrieve item availability information.

The getItemAvailability operation does not support the wildcard (asterisk (*)) for search criteria.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes item availability information. This table includes information about the business service properties that the getItemAvailability web service operation uses:

Group	Key	Description	Default Value
J4100001	J4100001_ERROR_PREFIX_1	Use this business service property to specify the prefix value the operation uses for error messages.	*Get Item Availability Sent In:*

Implementation Details

The following table includes information that can help determine whether the getItemAvailability operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • businessUnit • transactionUnitOfMeasure • quantityAvailable • itemId

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

insertInventoryItemStaging

This section provides an overview of the insertInventoryItemStaging web service operation.

Understanding the insertInventoryItemStaging Web Service Operation

The insertInventoryItemStaging web service operation is a database insert operation that enables consumers to insert one to many inventory item records into the staging table in the JD Edwards EnterpriseOne database.

The insertInventoryItemStaging web service operation uses auto commit transaction processing. Since inventory item records are autonomous, each inventory item staging record is immediately added to the F4101 Item Master Unedited Transaction Table (F4101Z1) and the entire group is not rolled back if an error occurs. If the operation is successful, the system inserts one to many records. You can specify the maximum number of records to insert using the Max Rows business service property.

If the operation fails, the system returns an error message to the consumer. The records that return an error will not be inserted into the database. However, records that do not error are committed to the database even if other records in the same transaction fail.

If all rows passed in were not inserted into the staging table, then the system returns an error message. The Max Rows business service property limits the number of inserted records. The consumer then sends the data in groups less than the Max Rows business service property or changes the Max Rows business service property to *0* (zero) so that all records are inserted.

Note. The data provided for insertion must be in the JD Edwards EnterpriseOne format. No formatting is performed before the insert is made.

Supported Functionality

This section discusses the functionality that the insertInventoryItemStaging operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The operation supports the insert of one to many inventory item records into the staging table in the JD Edwards EnterpriseOne database.

This web service operation does not support changing or deleting inventory item staging records.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes inventory item staging records. This table includes information about the business service properties that the insertInventoryItemStaging web service operation uses:

Group	Key	Description	Default Value
J4100006	J4100006_MAX_ROWS	Use this business service property to define the maximum number of rows that the operation inserts in the JD Edward EnterpriseOne database.	*999*
J4100006	J4100006_ITEM_ STOCKING_TYPE_CODE	Use this business service property to specify the stocking type code that the operation uses for the insert if the code is not provided.	*S*

Note. The Max Rows business service property limits the number of inserted records. Either send the data in groups less than the Max Rows business service property or change the Max Rows service constant to *0* (zero). If you set this value to *0,* the system inserts all records.

Implementation Details

The following table includes information that can help determine whether the insertInventoryItemStaging operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer that includes the number of rows that were inserted into the F4101Z1 table.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	No, records that return an error are not inserted into the database. Records that do not return an error are committed to the database even if other records in the same transaction fail. If you encounter errors during processing, review your insertion criteria and business service property settings and try your insert again.
Does this operation use record reservation?	No. This operation does not reserve records.

CHAPTER 14

Procurement Manager Web Service

This chapter provides an overview of the Procurement Manager web service, lists prerequisites, and discusses these web service operations:

- processPurchaseOrder
- processPurchaseOrderAcknowledge
- getPurchaseOrder

Understanding the Procurement Manager Web Service

The ProcurementManager web service (JP430000) manages the processing of procurement-related web service operations. This table includes a description of the procurement web service operations:

Operation	Description
processPurchaseOrder (J4300010)	Use this operation to add, change or cancel a purchase order within the JD Edwards EnterpriseOne Procurement system.
processPurchaseOrderAcknowledge (J4300020)	Use this operation to synchronize sales orders in the source system to the corresponding purchase orders in the JD Edwards EnterpriseOne Procurement system.
getPurchaseOrder (J4300030)	Use this operation to query the JD Edwards EnterpriseOne database for existing purchase order information.

Accessing Javadoc

To access Javadoc for the Procurement Manager web service and its related operations, review these Javadoc packages:

- JP430000 (ProcurementManager)
- J4300010 (processPurchaseOrder)
- J4300020 (processPurchaseOrderAcknowledge)
- J4300030 (getPurchaseOrder)

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before using the Procurement Manager web service, or any of the related web service operations, you must install and configure the JD Edwards EnterpriseOne Procurement system.

See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Procurement Management".

processPurchaseOrder

This section provides an overview of the processPurchaseOrder web service operation.

Understanding the processPurchaseOrder Web Service Operation

The processPurchaseOrder web service operation is an inbound transaction operation that enables consumers to process purchase order information within the JD Edwards EnterpriseOne system. The consumer can add, change or cancel a purchase order in the JD Edwards EnterpriseOne Procurement system.

If the operation is successful, the system returns a confirmation message which includes purchase order data to the consumer. If the operation fails, the system returns an error message to the consumer.

Supported Functionality

This section discusses the functionality that the processPurchaseOrder operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The processPurchaseOrder operation supports add, change or cancel of purchase orders, blanket orders, quote orders, requisition orders, and subcontracts. In Change mode the processPurchaseOrder operation supports add, change or cancel of detail lines for purchase orders, blanket orders, quote orders, requisition orders, and subcontracts.

The processPurchaseOrder web service operation does not support:

- Requisition self service.
- Purchase order generation.
- Suffix input at the header level.

  Although the suffix field is supported at the header level and is part of the key in JD Edwards World, the value in the suffix field for purchase orders in JD Edwards EnterpriseOne is always *000*. By preventing input of the suffix at the header level, the integration may not be compatible when integrating with JD Edwards World.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes purchase order information. This table includes information about the business service properties that the processPurchaseOrder web service operation uses:

Group	Key	Description	Default Value
J4300010	J4300010_PO_MBF_VERSION	Use this business service property to specify which version of the Purchase Orders program (P4310) the operation uses. See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Entering Purchase Orders," Setting Processing Options for Purchase Orders (P4310).	*ZJDE0001*
J4300010	J4300010_BYPASS_BSFN_WARNINGS	Use this business service property to specify whether the system ignores warnings during processing.	*1* **Note.** The system does not treat warnings as errors so the system keeps processing.
J4300010	J4300010_PREFIX_1	Use this business service property to specify the prefix value the operation uses for error messages on purchase order detail lines.	*Line No. Sent in:*

Implementation Details

The following table includes information that can help determine whether the processPurchaseOrder operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentCompany • documentTypeCode • documentSuffix • purchaseOrderLineNumber If the operation completes successfully, the system returns the confirm value object and no exception is thrown.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	This operation calls the PurchaseOrderInit business function (B4305320). This function reserves records in the JD Edwards EnterpriseOne system when the operation is changing or deleting existing data.

processPurchaseOrderAcknowledge

This section provides an overview of the processPurchaseOrderAcknowledge web service operation and lists prerequisites.

Understanding the processPurchaseOrderAcknowledge Web Service Operation

The processPurchaseOrderAcknowledge web service operation is an inbound transaction operation that enables consumers to process purchase order acknowledgements within the JD Edwards EnterpriseOne system. The consumer can synchronize sales orders in the source system to the corresponding purchase orders in the JD Edwards EnterpriseOne Procurement system.

If the operation is successful, the system returns a confirmation message which includes purchase order acknowledgement data to the consumer. If the operation fails, the system returns an error message to the consumer.

If the operation is successful, the system returns purchase order header and line keys to the caller through the ConfirmProcurementOrderAcknowledge web service operation. The system returns exceptions to the caller. The minimum required fields to acknowledge a purchase order in JD Edwards EnterpriseOne are:

- Header
  - statusOrderCode
  - purchaseOrderKey
- Detail (optional)
  - statusOrderCode
  - purchaseOrderLineKey

The processPurchaseOrderAcknowledge operation enables the consumer to specify the statusOrderCode at the header and detail level. The statusOrderCode indicates the type of operation that is being performed on the purchase order and is required. The status order codes are:

1: Accept

2: Accept with change

3: Pending

4: Product already shipped

5: Reject

This table includes information about the JD Edwards EnterpriseOne purchase order acknowledgement scenarios used by the processPurchaseOrderAcknowledge operation:

Status Order Code in Header	Status Order Code in Detail	Comment	Result
1	1	Accept	Only the detail status is changed and no data is updated.
1	2	Accept	The detail status is changed and other data is updated.
2	1	Accept	Only the detail status is changed and the data in the header is updated.
2	2	Accept with change	The detail status is changed, and the header and detail data is updated.
3	3	Pending	The status is changed to Pending as set in the processing options.
4	4	Product already shipped	There is no change in the EnterpriseOne system.
5	1, 2, 3, 4, 5	Reject	All the lines are rejected.

Status Order Code in Header	Status Order Code in Detail	Comment	Result
1	3		The header is not updated, the status is changed to Pending as set in the processing options, and the VR01 field in the detail is changed.
2	3		The header data is updated, the status is changed to Pending as set in the processing options, and the VR01 field in the detail is changed.
2	4		The header data is updated but no fields in the detail are updated.
2	5		The header data is updated and the status is changed.
3	2		The header is not updated, the detail status is changed, and other data is updated.
4	5		The data in the header is not updated and the status fields are changed.

Supported Functionality

This section discusses the functionality that the processPurchaseOrderAcknowledge operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The processPurchaseOrderAcknowledge operation enables a purchase order to be acknowledged, acknowledged and changed, appended, or rejected. The operation also supports sales order line splits. When the operation sends line split information to the JD Edwards EnterpriseOne system, it must be sent as an acknowledge with change so that a line is added to the purchase order. Two lines must be sent to split a line, both as an acknowledge with change. If a related sales order is not added, the operation performs a call in rejection mode and the corresponding JD Edwards EnterpriseOne purchase order is cancelled.

The processPurchaseOrderAcknowledge web service operation does not allow the status of cancelled lines to change.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes purchase order acknowledgment information. This table includes information about the business service properties used by the processPurchaseOrderAcknowledge operation:

Group	Key	Description	Default Value
J4300020	J4300020_PO_MBF_VERSION	Use this business service property to specify which version of the Purchase Orders program (P4310) the operation uses. See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Entering Purchase Orders," Setting Processing Options for Purchase Orders (P4310).	*ZJDE0001*
J4300020	J4300020_BYPASS_BSFN_WARNINGS	Use this business service property to specify whether the system ignores warnings during processing.	*1* The system does not treat warnings as errors so the system keeps processing.
J4300020	J4300020_PREFIX_1	Use this business service property to specify the prefix value the operation uses for error messages on purchase order detail lines.	*Line No. Sent in:*

Implementation Details

The following table includes information that can help determine whether the processPurchaseOrderAcknowledge operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentCompany • documentTypeCode • documentLineNumber • documentSuffix If the operation completes successfully, the system returns the confirm value object and no exception is thrown.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	This operation calls the PurchaseOrderAcknowledgeNotify business function (B4302190). This function reserves records in the JD Edwards EnterpriseOne system when the operation synchronizes sales orders in the source system with their corresponding purchase orders in the JD Edwards EnterpriseOne Procurement system.

Prerequisites

Before using the processPurchaseOrderAcknowledge web service operation, you must:

- Set the 3. Acknowledged Order Status Code, 4. Acknowledged With Change Order Status Code, and 5. Pending Order Status Code processing options on the Order Statuses tab of the Purchase Order XPI program (P43XPI).

  See Chapter 18, "Voucher Match Manager Web Service," Setting Processing Options for Purchase Order XPI (P43XPI), page 174.

- Set the Order Type to *OP* and Line Type for stock, non-stock or service lines in the Order Activity Rules from the Procurement System Setup menu (G43A41).

  See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Setting Up the Procurement System," Setting Up Order Activity Rules.

getPurchaseOrder

This section provides an overview of the getPurchaseOrder web service operation.

Understanding the getPurchaseOrder Web Service Operation

The getPurchaseOrder web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne Procurement system to retrieve existing purchase order information.

If the operation is successful, the system returns zero to many records to the consumer. You can specify the maximum number of records to return during a query using the Max Rows business service property. If the operation fails, the system returns an error message to the consumer.

The database query operation uses the V4301XPI business view which returns only the fields from the Purchase Order Header (F4301) and Purchase Order Detail File (F4311) tables that are needed for processing. Use of the business view allows retrieval of all required fields in a single optimized call.

Field retrieval is based on the selection criteria specified in the value object. The query published value object contains fields which filter the purchase order information retrieved by the JD Edwards EnterpriseOne operation.

This flow supports querying purchase order information in JD Edwards EnterpriseOne. The getPurchaseOrder web service operation enables source systems to query JD Edwards EnterpriseOne purchase order information in a real-time fashion. . The selectPurchaseOrder adapter service is invoked to retrieve purchase order information from the V4301XPI business view in JD Edwards EnterpriseOne based on the selection criteria specified in the GetPurchaseOrder published interface document. Retrieved results are grouped together with header and detail information and mapped to the ShowPurchaseOrder published interface document and delivered back to the source application.

The query published interface contains fields that can be used to filter the purchase order information that is retrieved. At minimum, any of the purchase order keys (documentNumber, documentCompany, or documentType) or the item (short item number), or any combination of these fields, is required to process the getPurchaseOrder integration.

Optional fields to include in the filter are:

- BusinessUnit
- CurrencyCode
- Buyer
- Supplier
- ShipTo
- PaymentTermsCode

Note. Preceding spaces are already padded to the businessUnit value in the getPurchaseOrder web service operation. The JD Edwards EnterpriseOne date format is used to query purchase orders based on dates in the getPurchaseOrder operation.

Supported Functionality

This section discusses the functionality that the getPurchaseOrder operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getPurchaseOrder web service operation enables consumers to query the JD Edwards EnterpriseOne Procurement system to retrieve existing purchase order information.

The getPurchaseOrder operation does not support the wildcard (asterisk (*)) for search criteria.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns when you query the JD Edwards EnterpriseOne database. This table includes information about the business service properties used by the getPurchaseOrder operation:

Group	Key	Description	Default Value
J4300030	J4300030_MAX_ROWS	Use this business service property to define the maximum number of rows that the operation returns when querying the JD Edwards EnterpriseOne database.	*100*

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0,* the system returns all matching records. Additionally, it is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database. The system returns an error if you perform a query with no selection criteria and this business service property is set to *0.*

Implementation Details

The following table includes information that can help determine whether the getPurchaseOrder operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany • orderSuffix • businessUnit • orderedBy • orderTakenBy • documentLineNumber • documentSuffix • itemId • itemProduct • itemCatalog If a business service exception is not thrown, then the operation completed successfully.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

CHAPTER 15

Purchase Order Receipt Manager Web Service

This chapter provides an overview of the Purchase Order Receipt Manager web service, lists a prerequisite, and discusses the processPurchaseOrderReceipt web service operation.

Understanding the Purchase Order Receipt Manager Web Service

The PurchaseOrderReceiptManager web service (JP43A000) manages the processing of purchase order receipt-related web service operations. This table includes a description of the purchase order receipt web service operation:

Operation	Description
processPurchaseOrderReceipt (J43A0010)	Use this operation to create partial and full purchase order receipts within the JD Edwards EnterpriseOne Procurement system. The operation supports basic and advanced serial number receipts.

Accessing Javadoc for the Purchase Order Receipt Manager Web Service Operations

To access Javadoc for the Purchase Order Receipt Manager web service and its related operations, review these Javadoc packages:

- JP43A000 (PurchaseOrderReceiptManager)
- J43A0010 (processPurchaseOrderReceipt)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisite

Before using the Purchase Order Receipt manager web service, or the related web service operation, you must install and configure the JD Edwards EnterpriseOne Procurement system.

See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Procurement Management".

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Processing Sales Orders," Understanding Serial Number Processing in the Distribution System.

processPurchaseOrderReceipt

This section provides an overview of the processPurchaseOrderReceipt web service operation.

Understanding the processPurchaseOrderReceipt Web Service Operation

The processPurchaseOrderReceipt web service operation is an inbound transaction operation that enables consumers to process purchase order receipt information within the JD Edwards EnterpriseOne system. The consumer can create full and partial purchase order receipts in the JD Edwards EnterpriseOne Procurement system. The operation also supports basic and advanced serial number receipts.

The processPurchaseOrderReceipt web service operation receives goods and services on a purchase order. The processPurchaseOrderReceipt operation calls the XPIProcessInboundReceipt business function (B4302240) to process the purchase order receipt. If there are no errors during processing, the business function may return information and warnings depending upon the scenario and report back to the processPurchaseOrderReceipt web service operation.

See Chapter 18, "Voucher Match Manager Web Service," Setting Processing Options for Purchase Order XPI (P43XPI), page 174.

Supported Functionality

This section discusses the functionality that the processPurchaseOrderReceipt operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

When creating purchase order receipts:

- Full and partial purchase order receipts are supported.
- Basic and advanced serial number receipts are supported.
- Receipt by purchase order and receipt by purchase order line item are supported.
- Receipt of one purchase order at a time is supported.

The processPurchaseOrderReceipt web service operation does not support the following functionality:

- Receipts of multiple purchase orders at a time.

- Update purchase order receipts.
- Receipts by account.
- Receive into multiple locations.
- Receipts for transportation.
- Receive and close purchase order and cancel purchase order receipt line.
- Receipt routing.
- Reverse receipt.
- Cascading receipts.
- Desktop receipt.
- Weight tag receipt (for wine industry).

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes purchase order information. This table includes information about the business service properties that the processPurchaseOrderReceipt web service operation uses:

Group	Key	Description	Default Value
J43A0010	J43A0010_PR_MBF_VERSION	Use this business service property to specify which version of the PO Receipts program (P4312) the operation uses. See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Using Receipt Processing," Setting Processing Options for PO Receipts (P4312).	*ZJDE0001*
J43A0010	J43A0010_PREFIX_1	Use this business service property to specify the prefix value the operation uses for error messages when JD Edwards EnterpriseOne runs the SerialNumberCacheProcess business function (B4302180) .	*Receipt Serial Number Sent in:*

Group	Key	Description	Default Value
J43A0010	J43A0010_PREFIX_2	Use this business service property to specify the prefix value the operation uses for error messages when JD Edwards EnterpriseOne runs the XPIProcessInboundReceipt business function (B4302240) for edit doc and edit line.	*Receipt Sent in:*
J43A0010	J43A0010_PREFIX_3	Use this business service property to specify the prefix value the operation uses for error messages when JD Edwards EnterpriseOne runs the XPIProcessInboundReceipt business function (B4302240) for end doc.	*Receipt Process WF Sent in:*

Implementation Details

The following table includes information that can help determine whether the processPurchaseOrderReceipt operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the web service call does not throw a business service exception, then the operation has completed successfully. There may be some warning messages in the returned value object which the consumer can review. If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany • receiptNumber • businessUnit • documentLineNumber • receiptLineNumber

Question	Answer
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. The consumer only needs to reverse the addition of records to the Item Location File (F41021) table in the event that a fatal server or connection error occurred when the rollback was occurring since the rollback of the F41021 table is handled manually with a business function call. All other transactions for the receipt are rolled back in the normal transaction processing handling.
Does this operation use record reservation?	This operation calls the XPIProcessInboundReceipt business function (B4302240) . This function reserves purchase order records in the JD Edwards EnterpriseOne system when the operation is processing receipts so that no other user can update the purchase order. If the purchase order for which the operation is creating a receipt is reserved by some other application, then the call to the B4302240 business function returns an error and stops any further processing.

CHAPTER 16

Sales Order Manager Web Service

This chapter provides an overview of the Sales Order Manager web service, lists prerequisites, and discusses these web service operations:

- processSalesOrder
- processSalesPriceAdjustment
- getItemPriceAndAvailability
- getCustomerItemPrice
- getSalesOrder
- getItemListPrice
- getSalesOrderPriceHistory

Understanding the Sales Order Manager Web Service

The SalesOrderManager web service (JP420000) manages the execution of sales order-related web service operations. This table includes a description of the sales order web service operations:

Operation	Description
processSalesOrder (J4200010)	Use this operation to process sales order information in the JD Edwards EnterpriseOne Sales Order Management system. This operation enables you to add, change, or cancel a sales order.
processSalesPriceAdjustment (J4200020)	Use this operation to process sales price adjustment data in the JD Edwards EnterpriseOne Advanced Pricing and Sales Order Management systems. This operation enables you to: • Add or update price adjustment definition records. • Add or update price adjustment detail records. • Add or update adjustment schedule records.
getItemPriceAndAvailability (J4200030)	Use this operation to retrieve item pricing and availability information from the JD Edwards EnterpriseOne Inventory Management system. **Note.** There are several operations that provide item pricing information to the consumer. Use this operation to review item availability information in addition to item pricing.

Operation	Description
getCustomerItemPrice (J4200040)	Use this operation to retrieve item pricing for a specified customer from the JD Edwards EnterpriseOne Inventory Management system. **Note.** There are several operations that provide item pricing information to the consumer. Use this operation to review item prices, including line-level discounts and markups, for specific customers.
getSalesOrder (J4200050)	Use this operation to review sales orders that exist in the JD Edwards EnterpriseOne Sales Order Management system.
getItemListPrice (J4200060)	Use this operation to retrieve a list of base prices from the JD Edwards EnterpriseOne Inventory Management system for a specified item. **Note.** There are several operations that provide item pricing information to the consumer. Use this operation to review base prices for specified items. This operation does not provide customer-specific prices or availability information.
getSalesOrderPriceHistory (J4200070)	Use this operation to retrieve historical pricing information from the JD Edwards EnterpriseOne Sales Order Management system.

Accessing Javadoc for the Sales Order Manager Web Service Operations

To access Javadoc for the Sales Order Manager web service and its related operations, review these Javadoc packages:

- JP420000 (SalesOrderManager)
- J4200010 (processSalesOrder)
- J4200020 (processSalesPriceAdjustment)
- J4200030 (getItemPriceAndAvailability)
- J4200040 (getCustomerItemPrice)
- J4200050 (getSalesOrder)
- J4200060 (getItemListPrice)
- J4200070 (getSalesOrderPriceHistory)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before you use the Sales Order Manager web service, or any of the related operations, you must install and set up these JD Edwards EnterpriseOne systems:

- Sales Order Management
- Inventory Management

Note. Many of these operations managed by the SalesOrderManager web service process pricing information. If you want these operations to process pricing information using advanced pricing functions, you must install and set up the JD Edwards EnterpriseOne Advanced Pricing system. If you do not set up this system, these operations will process, but they will not include advanced pricing calculations.

See *JD Edwards EnterpriseOne Advanced Pricing 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Advanced Pricing".

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Sales Order Management".

See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Inventory Management".

processSalesOrder

This section provides an overview of the processSalesOrder web service operation and lists prerequisites.

Understanding the processSalesOrder Web Service Operation

The processSalesOrder web service operation is an inbound transaction operation that enables consumers to add, change, and cancel these order types in the JD Edwards EnterpriseOne Sales Order Management system:

- Sales orders
- Blanket orders
- Credit orders
- Direct ship orders
- Quote orders
- Transfer orders
- Transportation orders

If the operation encounters errors during processing, the system returns those errors to the consumer.

Supported Functionality

This section discusses the functionality that the processSalesOrder operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The processSalesOrder operation supports inventory/stock items, configured items, non-stock items such as miscellaneous, text lines and freight, and direct ship items. An order may contain one or more of these line types.

Blanket Orders

Though the solution does support the addition, change or cancellation of blanket orders, the solution does not support automatic blanket order release. Additionally, the solution does not support blanket orders for configured items.

Advanced Pricing

The solution supports advanced pricing for line, basket, and order-level pricing. The operation does not support advanced pricing when an override price is passed in by the consumer.

Kit Items

The solution does not support kit items. If the consumer enters a kit item, the system will process the information, but kit processing is not initiated in the JD Edwards EnterpriseOne system. The operation returns a*Kit Components Are Not Processed* message to the consumer.

If a kit parent item is entered by the consumer, the system processes the parent, but does not initiate kit processing for components. To resolve the issue, the consumer must use the JD Edwards EnterpriseOne Sales Order Entry program (P4210) to choose the kits components.

Configured Items

The solution does support the processing of configured items. The operation provides the consumer with a field structure that enables the specification of configured items and their segments.

Configured items may be pre-evaluated and passed into the JD Edwards EnterpriseOne system as they are, which is known as *bypass*. Alternatively, consumers can pass items into the JD Edwards EnterpriseOne system and have the item evaluated using JD Edwards EnterpriseOne defined rules and logic.

Configured parent items may contain child items of varying types, such as stock, non-stock and configurable.

Configured items are enabled for sales orders and quote orders only at the configured item parent level.

The solution does not support blanket orders for configured items.

When processing configured items, be aware of the following items:

- The stocking type for a configured item is hard-coded as *C*.
- The consumer must provide segment answers when adding or changing a configured item.
- To utilize the JD Edwards EnterpriseOne rules and logic for evaluation, segment answers and configured components are required when adding or changing a configured item.
- To bypass evaluation in the JD Edwards EnterpriseOne system, the consumer must provide the segment answers and all components when adding or changing a configured item.
- The solution does not validate segment answers and cross-segment editing rules.
- The consumer must sequence the configuration by children before siblings.

For example, configured item A has components A1 and A2. Component A1 has another child component, A11. Using this example, the input to the operation should follow this sequence:

1. parentItem = null, childItem = A
2. parentItem = A, childItem = A1
3. parentItem = A, childItem = A2
4. parentItem = A1, childItem = A11

- The solution does not support the changing or cancellation of individual components of a configured item.
- When processing the cancellation of a configured item, the operation also cancels the components from the sales order details.
- The solution does not support back orders, transfer orders, direct ship orders, and inter-branch orders for configured items.
- The price will be overridden when the isZeroPriceOverride field is set to *true*, or there is a price (domestic/foreign) in the input document.
- The solution does not support:
  - Dynamic rules validation during configuration.
  - The limitation of configured selections to include only valid responses.
  - Calculation of pricing rules based on a user-defined date.

Setup Considerations

Before using this operation, you can set business service properties to specify how the system processes sales order information. This table includes information about the business service properties that the processSalesOrder web service operation uses:

Group	Key	Description	Default Value
J4200010	J4200010_SOE_MBF_ VERSION	Use this business service property to specify the version of the Sales Order Entry program (P4210) that the operation uses when processing data.	ZJDE0001
J4200010	J4200010_BYPASS_ WARNINGS	Use this business service property to specify whether the operation bypasses warnings that occur during processing. If you choose to bypass warnings, the operation continues processing to the end, and returns all warnings to the consumer. If you choose not to bypass warnings, change the default value to *2*. Using this setting, the operation stops processing when a warning is encountered and the warning is returned to the consumer.	*1:* Bypass Warnings

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

See Also

JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide, "Entering Sales Orders," Setting Processing Options for Sales Order Entry (P4210)

Implementation Details

The following table includes information that can help determine whether the processSalesOrder operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany • businessUnit • entity – soldTo • itemId • documentLineNumber
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. In the event that the operation encounters errors, all JD Edwards EnterpriseOne tables are returned to their original state, and no updates are performed. No manual updates are necessary.
Does this operation use record reservation?	This operation reserves records in the JD Edwards EnterpriseOne system during the change or cancellation of a sales order.

Prerequisites

Before using this operation, you must be familiar with entering, updating and deleting sales orders in the JD Edwards EnterpriseOne system.

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Entering Sales Orders," Entering Sales Order Header and Detail Information.

processSalesPriceAdjustment

This section provides an overview of the processSalesPriceAdjustment web service operation, lists a prerequisite, and discusses how to set processing options for Live Promotions Interface Processing (R45720Z).

Understanding the processSalesPriceAdjustment Web Service Operation

The processSalesPriceAdjustment operation is an inbound transaction operation that enables consumers to process sales price adjustment information within the JD Edwards EnterpriseOne system.

The consumer can add this information to the JD Edwards EnterpriseOne Sales Order Management system:

- Price adjustment definition records.
- Price adjustment detail records.
- Price adjustment schedule records.

Consumers are also able to modify existing price adjustment detail records, provided that values for the adjustment name and adjustment ID are passed in.

If the system encounters errors while processing transactions using this operation, error messages are generated and returned to the consumer.

Be aware that this operation updates only the pricing setup records in the JD Edwards EnterpriseOne system. Existing sales orders are not updated with pricing details that are entered into the system using this operation. To reprice existing sales orders using the updated pricing information, you must process the Sales Order Batch Price/Cost Update program (R42950).

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Updating Prices," Updating Prices for a Customer.

Note. To add a price adjustment detail record, you must include a unit of measure (UoM). If you do not include a unit of measure, the system returns a hard error to the consumer.

If currency processing is turned on, you must include a currency code. If no currency code is passed in, and currency processing is enabled, the system returns a hard error to the consumer.

If no dates are passed in, the system uses the system date as the effective date, and an expiration date that is generated by business function B4000630.

Supported Functionality

This section discusses the functionality that the processSalesPriceAdjustment operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

When adding price adjustment definition records:

- Item price groups, customer price groups, and order detail groups are not supported.
- Basket, order, and volume level adjustments are not supported.

  Only line level adjustments are supported.
- Weight, amount, and quality level breaks are not supported.

  Only quantity level breaks are supported.
- These adjustment control codes are supported:
  - Blank
  - *2:* Print on Invoice
  - *4:* Accruals
- When making changes to price adjustment detail records:
  - The From Level value always starts at *1*.

- Basis code *5* (Add on Amount) is supported.

Setup Considerations

Before you use this operation, you can set a business service property to define which version of the Live Promotions Interface Processing program (R45720Z) to use. This table lists the business service properties that the processSalesPriceAdjustment operation uses:

Group	Key	Description	Default Value
J4200020	J4200020_PROM_MBF_VERSION	Use this business service property to specify the version of the Live Promotions Interface Processing program (R45720Z) that this operation uses.	ZJDE0002

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

The following table includes information that can help determine whether the processSalesPriceAdjustment operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • adjustmentTypeCode • priceAdjustmentId • unitOfMeasureTransaction • actionType • processingVersion
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	No. This operation does not reserve records in the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before using the processSalesPriceAdjustment operation, you must install and configure the JD Edwards EnterpriseOne Advanced Pricing system.

See *JD Edwards EnterpriseOne Advanced Pricing 9.0 Implementation Guide*, "Setting Up EnterpriseOne Advanced Pricing".

Setting Processing Options for Live Promotions Interface Processing (R45720Z)

You use processing options to set up default values and processing data for a program.

Preference Hierarchy

1. Enter the preference hierarchy name to be used. (Required)

Use this processing option to specify a preference type or a price adjustment hierarchy defined in user-defined code (UDC) table 40/PR. This value gets stored in the Price Adjustment Type table (F4071) in field ATPRFR (Preference) and its associated hierarchy is used for resolving pricing adjustments.

Adjustment Definition

The processing options on this tab are used to define the adjustment definition. One of the input parameters, such as costTypeCode, determines whether the adjustment is a Bill Back or an Off Invoice adjustment. The value sent in becomes part of the adjustment definition record that is stored in the F4071 table. Adjustment types include 1(Bill Back: Adjustment Control Code = 4, Accruals) or 2 (Off Invoice: Adjustment Control Code = 2, Print on Invoice). If no value is passed in for costTypeCode field, Adjustment Control Code, stored in field ATACNT of the F4071 table, is blank.

1. Enter GL Offset for Bill Back adjustment definition.

Use this processing option to specify the trade account that the system uses as the offset when you post invoices or vouchers. This value gets stored in the Price Adjustment Type table (F4071) in field ATGLC (GLClass).

2. Enter Subledger for Bill Back adjustment definition.

Use this processing option to identify a detailed, auxiliary account within a general ledger account. A subledger can be an equipment item number or an address book number. If you enter a subledger, you must also specify the subledger type.

This value is stored in the Price Adjustment Type table (F4071) in field ATSBIF (SubledgerInformation).

3. Enter GL Offset for Off Invoice adjustment definition.

Use this processing option to specify the trade account that the system uses as the offset when you post invoices or vouchers. This value gets stored in the Price Adjustment Type table (F4071) in field ATGLC (GLClass).

4. Enter Subledger for Off Invoice adjustment definition.

Use this processing option to identify a detailed, auxiliary account within a general ledger account. A subledger can be an equipment item number or an address book number. If you enter a subledger, you must also specify the subledger type.

This value is stored in the Price Adjustment Type table (F4071) in field ATSBIF (SubledgerInformation).

Schedule Option

1. Enter Schedule Option. Use this processing option to specify the schedule option. Values are:

1: Single

2: Customer

3: Manual

Single Schedule Option

1. Enter schedule name. (Required) Use this processing option to specify the name of the schedule.

2. Enter prefix (1 char) for adjustment name. (Required) Use this processing option to specify the one-character prefix that is used for the adjustment name. The adjustment name is the prefix plus the next number plus the suffix for Bill Back and Off Invoice adjustment names. For example, if the prefix is *S* and the suffix is *B* (Bill Back), the adjustment name might be S123456B.

3. Enter suffix (1 char) for adjustment name. (Required) Use this processing option to specify the one-character suffix that is used for the adjustment name. The adjustment name is the prefix plus the next number plus the suffix for Bill Back and Off Invoice adjustment names. For example, if the prefix is *S* and the suffix is *B* (Bill Back), the adjustment name might be S123456B.

4. Enter increment for sequence in schedule. If blank, 1.0 will be used. Use this processing option to specify the numeric increment for sequence in schedule. If you leave this option blank, the system uses an increment of 1.0.

5. Enter start reserved range of sequence numbers for promotions. Use this processing option to specify the beginning value in a range of sequence numbers that are reserved for promotions.

6. Enter end reserved range of sequence numbers for promotions. Use this processing option to specify the ending value in a range of sequence numbers that are reserved for promotions.

Customer Schedule Option

1. Enter prefix (1 char) for adjustment name. (Required) Use this processing option to specify the one-character prefix that is used for the adjustment name. The adjustment name is the prefix plus the next number plus the suffix for Bill Back and Off Invoice adjustment names. For example, if the prefix is *S* and the suffix is *B* (Bill Back), the adjustment name might be S123456B.

2. Enter suffix (1 char) for adjustment name. (Required) Use this processing option to specify the one-character suffix that is used for the adjustment name. The adjustment name is the prefix plus the next number plus the suffix for Bill Back and Off Invoice adjustment names. For example, if the prefix is *S* and the suffix is *B* (Bill Back), the adjustment name might be S123456B.

3. Enter increment for sequence in schedule. If blank, 1.0 will be used. Use this processing option to specify the numeric increment for sequence in schedule. If you leave this option blank, the system uses an increment of 1.0.

4. Enter start reserved range of sequence numbers for promotions.

Use this processing option to specify the beginning value in a range of sequence numbers that are reserved for promotions.

5. Enter end reserved range of sequence numbers for promotions.

Use this processing option to specify the ending value in a range of sequence numbers that are reserved for promotions.

Manual Schedule Maintenance Option

1. Enter Adjustment name for Bill Back spending method.

Use this processing option to specify the adjustment name for Bill Back spending method.

Note. If you leave options 3 and 4 blank, you must enter a value in this processing option and in option 2.

2. Enter Adjustment name for Off Invoice spending method.

Use this processing option to specify the adjustment name for Off Invoice spending method.

Note. If you leave options 3 and 4 blank, you must enter a value in this processing option and in option 1.

3. Enter prefix (1 char) for adjustment name.

Use this processing option to specify the one-character prefix that is used for the adjustment name. The adjustment name is the prefix plus the next number plus the suffix for Bill Back and Off Invoice adjustment names. For example, if the prefix is *S* and the suffix is *B* (Bill Back), the adjustment name might be S123456B.

Note. If you leave options 1 and 2 blank, you must enter a value in this option and in option 4.

4. Enter suffix (1 char) for adjustment name.

Use this processing option to specify the one-character suffix that is used for the adjustment name. The adjustment name is the prefix plus the next number plus the suffix for Bill Back and Off Invoice adjustment names. For example, if the prefix is *S* and the suffix is *B* (Bill Back), the adjustment name might be S123456B.

Note. If you leave options 1 and 2 blank, you must enter a value in this option and in option 3.

getItemPriceAndAvailability

This section provides an overview of the getItemPriceAndAvailability web service operation and lists a prerequisite.

Understanding the getItemPriceAndAvailability Web Service Operation

The getItemPriceAndAvailability web service operation is an inbound transaction operation that enables consumers to retrieve pricing, availability, and branch/plant information for items that are stored in the JD Edwards EnterpriseOne Inventory Management system.

Note. There are several operations that return item prices to the consumer. Use this operation if you want to review item availability information in addition to pricing information.

This operation uses the information that the consumer passes in, along with the Price and Availability Header business function (B4204100), to simulate the creation of a sales order header and sales order detail information. The system then uses this simulated sales order to retrieve pricing information. The price is based on the branch/plant that the consumer passes in. If the consumer does not pass in a branch/plant, the system calculates the price using the branch/plant from the processing options of the P4210. The operation then returns to the consumer the unit price and extended price for the item.

Additionally, the operation uses the Retrieve F41021 Records business function (B4204120) to calculate the availability of the item in the specified branch/plant. The operation then returns to the consumer the availability information, along with the address book number and mailing address of the branch/plant.

If the operation encounters errors or warnings during processing, the system returns those errors and warnings to the consumer.

Supported Functionality

This section discusses the functionality that the getItemPriceAndAvailability operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

This operation does not support configured items, as configured items typically do not have availability information.

If you are using advanced pricing, the operation applies markups and discounts. The operation does not apply free goods and rebates.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves and calculates price and availability information. This table includes information about the business service properties that the getItemPriceAndAvailability operation uses:

Group	Key	Description	Default Value
J4200030	J4200030_SOE_MBF_VERSION	Use this business service property to specify the version of the Sales Order Entry program (P4210) that the operation uses when processing price and availability information.	ZJDE0001
J4200030	J4200030_BYPASS_BSFN_WARNINGS	Use this business service property to specify whether the operation bypasses warnings that occur during processing. If you choose to bypass warnings, the operation continues processing to the end, and returns all warnings to the consumer. If you choose not to bypass warnings, change the default value to *2*. Using this setting, the operation stops processing when a warning is encountered and the warning is returned to the consumer.	*1:* Bypass Warnings

The getItemPriceAndAvailability operation uses the Sales Order Entry program (P4210) to simulate the creation of a sales order for the purposes of retrieving price and availability information. Typically, the version of the Sales Order Entry program that is used to create and process sales orders is set up to include full editing and validations. The getItemPriceAndAvailability operation does not need to complete these processes. Therefore, to improve performance, you might consider using a different version of the Sales Order Entry program when processing this operation.

In addition to using a separate version of the P4210, it is also recommended that you use these guidelines when setting the processing options for the version used by this operation:

- Order Holds must be blank.
- Activate Availability Checking must be blank.
- Blanket or Quote Processing must be blank.
- Inventory Commitment Preferences must be blank.
- If Preferences are turned on, the following preferences must not be used:
  - Product Allocation
  - Grade and Potency
  - Next Order Status
  - Sales Commissions

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Entering Sales Orders," Setting Processing Options for Sales Order Entry (P4210).

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getItemPriceAndAvailability operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system does not return errors to the consumer. In some instances, the consumer does not receive a return message. For example, if the item in the query does not have price or availability information, the operation does not return data to the consumer, even though it has successfully processed. If the operation is successful in finding matching data, price and availability information for the items included in the query are returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • priceUnit • priceExtended • quantityAvailable
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a query operation. No transactions occur during the processing of this operation. Therefore, no data updates are necessary if errors are encountered.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisite

Before you can process this operation successfully, you must enter items into the JD Edwards EnterpriseOne Inventory Management system.

See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Setting Up the Inventory Management System" and *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Entering Item Information".

getCustomerItemPrice

This section provides an overview of the getCustomerItemPrice web service operation and lists prerequisites.

Understanding the getCustomerItemPrice Web Service Operation

The getCustomerItemPrice web service operation is an inbound transaction operation that enables consumers to retrieve item pricing information that is based on the base pricing or on advanced pricing with line-level discounts or markups.

Note. There are several operations that return item prices to consumers. Use this operation if you want to review the cost of an item for a particular customer.

This operation uses the information that the consumer passes in, along with the Price and Availability Header business function (B4204100), to simulate the creation of a sales order header and sales order detail information. The system then uses this simulated sales order to retrieve pricing information for the specified item and customer.

You can include only one item in each request.

Supported Functionality

This section discusses the functionality that the getCustomerItemPrice operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The getCustomerItemPrice operation does not include basket or order-level adjustments when calculating the final price of the item.

Consumers can request pricing information for kit and configured items. However, the operation returns prices only for the parent item if one has been specified.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system retrieves and calculates price and availability information. This table includes information about the business service properties that the getCustomerItemPrice operation uses:

Group	Key	Description	Default Value
J4200030	J4200030_SOE_MBF_VERSION	Use this business service property to specify the version of the Sales Order Entry program (P4210) that the operation uses when processing price and availability information.	ZJDE0001
J4200030	J4200030_BYPASS_BSFN_WARNINGS	Use this business service property to specify whether the operation bypasses warnings that occur during processing. If you choose to bypass warnings, the operation continues processing to the end, and returns all warnings to the consumer. If you choose not to bypass warnings, change the default value to *2*. Using this setting, the operation stops processing when a warning is encountered and the warning is returned to the consumer.	*1:* Bypass Warnings

The getCustomerItemPrice operation uses the Sales Order Entry program (P4210) to simulate the creation of a sales order for the purposes of retrieving price and availability information. Typically, the version of the Sales Order Entry program that is used to create and process sales orders is set up to include full editing and validations. The getCustomerItemPrice operation does not need to complete these processes. Therefore, to improve performance, you might consider using a different version of the Sales Order Entry program when processing this operation.

In addition to using a separate version of the P4210, it is also recommended that you use these guidelines when setting the processing options for the version used by this operation:

- Order Holds must be blank.
- Activate Availability Checking must be blank.
- Blanket or Quote Processing must be blank.
- Inventory Commitment Preferences must be blank.
- If Preferences are turned on, the following preferences must not be used:
  - Product Allocation
  - Grade and Potency
  - Next Order Status
  - Sales Commissions

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Entering Sales Orders," Setting Processing Options for Sales Order Entry (P4210).

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getCustomerItemPrice operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system does not return errors to the consumer. In some instances, the consumer does not receive a return message. For example, if the item in the query does not have price or availability information, the operation does not return data to the consumer, even though it has successfully processed. If the operation is successful in finding matching data, price and availability information for the items included in the query are returned to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • priceUnitDomestic • priceUnitForeign • priceExtendedDomestic • priceExtendedForeign • costUnitDomestic • costUnitForeign • costExtendedDomestic • costExtendedForeign
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a query operation. No transactions occur during the processing of this operation. Therefore, no data updates are necessary if errors are encountered.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisites

Before you can use this operation, you must enter item and pricing information into the JD Edwards EnterpriseOne Inventory Management system.

See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Entering Item Information".

getSalesOrder

This section provides an overview of the getSalesOrder web service operation and lists a prerequisite.

Understanding the getSalesOrder Web Service Operation

The getSalesOrder web service operation is a database query operation that enables consumers to review sales orders that exist in the JD Edwards EnterpriseOne Sales Order Management system.

The getSalesOrder operation uses the Sales Order Inquiry business view (V4211XBP) to select records from the JD Edwards EnterpriseOne Sales Order Management system based on the information that the consumer enters. If no errors are encountered, the operation searches the database for records that match the consumer's query, adding those records to the result set.

The operation returns one or more sales order header records to the consumer, based on the information that was passed in. In addition, the operation returns all sales order detail lines associated with those sales order header records.

If the operation encounters errors while querying the database, it stops processing and returns the errors to the consumer.

Note. The consumer must enter search criteria in order for the operation to process successfully. If the consumer attempts to query the database without specifying any search criteria, the operation returns an error to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns when you query the JD Edwards EnterpriseOne database. This table includes information about the business service properties that the getSalesOrder operation uses:

Group	Key	Description	Default Value
J4200050	J4200050_MAX_ROWS	Use this business service property to specify the maximum number of rows that the query will return. **Note.** The number of rows includes both sales order header records and sales order detail records.	*0:* Return all records.

Note. If you leave this constant set to *0*, the system returns all records that match the specified search criteria. To improve processing time, it is recommended that you set the default value for this business service property to a value other than *0*.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help you determine whether the getSalesOrder operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. The records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany • Entity – Sold To • Entity – Ship To
If I encounter errors while processing the operation, do I need to reverse the transaction?	This is a query operation. No transactions occur during the processing of this operation. Therefore, no data updates are necessary if errors are encountered.
Does this operation use record reservation.	No. The operation does not reserve records that are returned in the query.

Prerequisite

Before you can use this operation, sales orders must exist in the JD Edwards EnterpriseOne database.

See *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Entering Sales Orders" and *JD Edwards EnterpriseOne Sales Order Management 9.0 Implementation Guide*, "Entering Sales Orders for Customer Service Representatives".

getItemListPrice

This section provides an overview of the getItemListPrice web service operation and lists prerequisites.

Understanding the getItemListPrice Web Service Operation

The getItemListPrice web service operation is a database query operation that enables consumers to view a list of base prices for a specified item. Consumers can also specify the business unit in order to view a list of base prices for an item by business unit.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns for a query. This table includes information about the business service properties that the getItemListPrice operation uses:

Group	Key	Description	Default Value
J4200060	J4200060_MAX_ROWS	Use this business service property to specify the maximum number of records that the system returns for a query.	*0:* Return All Records

Note. If you leave this constant set to *0*, the system returns all records that match the specified search criteria. To improve processing time, it is recommended that you set the default value for this business service property to a value other than *0*.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getItemListPrice operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a list of prices for the item and or business unit specified by the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for the priceList field.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a query operation. No transactions occur during the processing of this operation. Therefore, no data updates are necessary if errors are encountered.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

Prerequisites

Before using this operation, you must enter items and base pricing information in the JD Edwards EnterpriseOne Inventory Management system.

See *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Entering Item Information," Entering Item Master Information and *JD Edwards EnterpriseOne Inventory Management 9.0 Implementation Guide*, "Entering Item Information," Entering Sales Price Information.

getSalesOrderPriceHistory

This section provides an overview of the getSalesOrderPriceHistory web service operation.

Understanding the getSalesOrderPriceHistory Web Service Operation

The getSalesOrderPriceHistory web service operation is a database query operation that enables consumers to inquire on price history information that resides in the JD Edwards EnterpriseOne Price Adjustment Ledger table (F4074). The consumer can search for price history by passing in any combination of these fields:

- Order Number
- Order Type
- Order Company
- Line Number

Note. You must pass in at least one of these fields when entering a query. If you enter a blank query, the system returns an error message.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns for a query. This table includes information about the business service properties that the getSalesOrderPriceHistory operation uses:

Group	Key	Description	Default Value
J4200070	J4200070_MAX_ROWS	Use this business service property to specify the maximum number of records that the system returns for a query.	*0:* Return all matching records.

Note. If you enter a default value of *0* for this business service property, the system does not limit the number of rows returned during a query. To improve processing time, it is recommended that you set the default value for this business service property to a value other than *0*.

See *JD Edwards EnterpriseOne 8.98 Business Services Development Methodology Guide, Creating a Business Service, Managing Business Service Properties*

Implementation Details

This table includes information that can help determine whether the getSalesOrderPriceHistory operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a list of price history records to the consumer. The records that are returned to the consumer include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • documentNumber • documentTypeCode • documentCompany • documentLineNumber
If I encounter errors while processing a transaction, do I need to reverse the transaction?	If you encounter errors while processing this operation, verify your business service properties and selection criteria. Then perform your query again. If the system does not return any matching records, verify that matching records exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. The operation does not reserve records within the JD Edwards EnterpriseOne system during processing.

CHAPTER 17

Supplier Manager Web Service

This chapter provides an overview of the Supplier Manager web service, lists prerequisites, and discusses these web service operations:

- processSupplier
- getSupplier

Understanding the Supplier Manager Web Service

The SupplierManager web service (JP010030) manages the processing of supplier-related web service operations. This table includes a description of the supplier manager web service operations:

Operation	Description
processSupplier (J0100031)	Use this operation to complete these tasks within the JD Edwards EnterpriseOne Address Book, JD Edwards EnterpriseOne Accounts Payable, and JD Edwards EnterpriseOne Accounts Receivable systems. • Add, change or delete supplier records. • Add, change or delete supplier address book records. • Add, change or delete supplier phone numbers. • Add, change or delete supplier electronic address records.
getSupplier (J0100032)	Use this operation to query the JD Edwards EnterpriseOne database for existing supplier information.

Accessing Javadoc for the Supplier Manager Web Service Operations

To access Javadoc for the Supplier Manager web service and its related operations, review these Javadoc packages:

- JP010030 (SupplierManager)
- J0100031 (processSupplier)
- J0100032 (getSupplier)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before using the Supplier Manager web service, or any of the related web service operations, you must install and configure the JD Edwards EnterpriseOne Address Book and JD Edwards EnterpriseOne Accounts Payable systems.

See *JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide*, "Getting Started With JD Edwards EnterpriseOne Address Book".

See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Getting Started with JD Edwards EnterpriseOne Accounts Payable".

processSupplier

This section provides an overview of the processSupplier web service operation.

Understanding the processSupplier Web Service Operation

The processSupplier web service operation is an inbound transaction operation that enables consumers to process supplier information within the JD Edwards EnterpriseOne system. The processSupplier web service operation enables source systems to process supplier information in JD Edwards EnterpriseOne in a real-time fashion. The web service uses JD Edwards EnterpriseOne master business functions to process the supplier information to add, change or delete supplier related records.

The consumer can complete these tasks in the JD Edwards EnterpriseOne Address Book, JD Edwards EnterpriseOne Accounts Payable, and JD Edwards EnterpriseOne Accounts Receivable systems:

- Add, change or delete supplier records.
- Add, change or delete address book records.
- Add, change or delete phone number records.
- Add, change or delete electronic address records.

After processing supplier information, the processSupplier web service operation calls the processAddressBook web service operation to process the address book record. If the operation encounters no errors then the processAddressBook web service operation calls the processPhones and processElectronicAddresses web service operations. The AddressBookProcessor web service operation will return either success messages or error messages if the transaction failed.

If AddressBookProcessor returns success, the processSupplier operation calls the SupplierMaster business function (N0100043) to process the supplier records regardless of the success or failure of the phone and electronic address processing.

Note. The processSupplier operation processes address book information and supplier records even if the processPhones and processElectronicAddresses web service operations fail. If the system encounters errors while processing the address book or supplier record information the system rolls back all transactions.

Every change in the supplier address book is a net change. If a null is passed in a variable field the system will check to see if that field has a value in the database. If the field does have a value in the database then the system will not overwrite the field with the null.

Supported Functionality

This section discusses the functionality that the processSupplier operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

When adding, changing or deleting supplier records, the processSupplier operation updates the Supplier Master (F0401), Address Book Master (F0101), Address Book -Who's Who (F0111), Address Book - Phone Numbers (F0115), Address by Date (F0116), Electronic Address (F01151), and Address Organization Structure Master (F0150) tables.

The processSupplier operation does not support localizations.

Setup Considerations

Before you use this operation, you can set business service properties to specify how the system processes supplier information. This table includes information about the business service properties that the processSupplier web service operation uses:

Group	Key	Description	Default Value
J0100031	J0100031_Supplier_MBF_VERSION	Use this business service property to specify which version of the Supplier Master MBF – PO program (P0100043) the operation uses. See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Entering Supplier Information," Setting Processing Options for Supplier Master MBF - PO (P0100043).	*ZJDE0001*
J0100001	J0100031_AB_MBF_VERSION	Use this business service property to specify which version of the Address Book MBF – PO (for N0100041) program (P0100041) the operation uses. See *JD Edwards EnterpriseOne Address Book 9.0 Implementation Guide*, "Entering Address Book Records," Setting Processing Options for Address Book MBF (P0100041).	*ZJDE0001*

Implementation Details

The following table includes information that can help determine whether the processSupplier operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • EntityId • EntityName • EntityTypeCode If the operation completes successfully, the AN8 (Address Number) and ALPH (Alpha Name) fields in the F0101 and F0111 tables and the AN8 field in the F0401 table will be populated.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. The processSupplier operation processes address book information and supplier records even if the processPhones and processElectronicAddresses operations fail. If the system encounters errors while processing the address book or supplier record information the system rolls back all transactions. No manual update is necessary.
Does this operation use record reservation?	No. This operation does not reserve records.

getSupplier

This section provides an overview of the getSupplier web service operation.

Understanding the getSupplier Web Service Operation

The getSupplier web service operation is a database query operation that enables consumers to query the JD Edwards EnterpriseOne database to retrieve existing supplier information, phone numbers, and electronic address records.

When the getSupplier web service operation is called, it first performs a select on the V0401XPI view (business view over the F0101, F0116, and F0401 tables) which returns zero or more supplier records. If the V0401XPI view produces errors then the program stops processing. Otherwise, the program loops through the returned supplier records and calls the getContact web service operation for each of the supplier records returned. Finally, the getSupplier operation exits and returns any errors to the Supplier Manager web service.

If the operation is successful, the system returns zero to many supplier records to the consumer. You can specify the maximum number of records to return during a query using the Max Rows business service property. It is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database. If the operation fails, the system returns an error message to the consumer.

Setup Considerations

Before you use this operation, you can set business service properties to specify how many records the system returns when you query the JD Edwards EnterpriseOne database. This table includes information about the business service properties used by the getSupplier operation:

Group	Key	Description	Default Value
J0100032	J0100032_MAX_ROWS	Use this business service property to define the maximum number of rows that the operation returns when querying the JD Edwards EnterpriseOne database.	*100*

Note. It is strongly recommended that you set this business service property to a value other than *0* (zero). If you leave this value set to *0,* the system returns all matching records. Additionally, it is recommended that you specify selection criteria when you query the JD Edwards EnterpriseOne database. However, open queries are allowed.

Implementation Details

The following table includes information that can help determine whether the getSupplier operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If the operation completes successfully, the system returns records that match your search criteria. These records include all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, if the query finds matching records, the system returns non-zero values for these fields: • MailingName • EntityId • EntityTypeCode Also, the operation completes successfully if no errors are returned.
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This is a database query operation that does not perform transactions. If you encounter errors during processing, review your search criteria and business service property settings and try your query again. If the operation does not return any records, verify that records matching your query exist in the JD Edwards EnterpriseOne database.
Does this operation use record reservation?	No. This operation does not reserve records that are returned in a query.

CHAPTER 18

Voucher Match Manager Web Service

This chapter provides an overview of the Voucher Match Manager web service, lists prerequisites, and discusses the processVoucherMatch web service operation.

Understanding the Voucher Match Manager Web Service

The VoucherMatchManager web service (JP43B000) manages the processing of voucher match-related web service operations. This table includes a description of the voucher match web service operations:

Operation	Description
processVoucherMatch (J43B0010)	Use this operation to add voucher match records for logged vouchers and both two-way and three-way voucher match within the JD Edwards EnterpriseOne Procurement system. This operation also supports additional charge information lines for a voucher along with purchase order lines being vouchered.

Accessing Javadoc for the Voucher Match Manager Web Service Operations

To access Javadoc for the Voucher Match Manager web service and its related operations, review these Javadoc packages:

- JP43B000 (VoucherMatchManager)
- J43B0010 (processVoucherMatch)

See Chapter 2, "Accessing Additional Information about Business Services," Accessing Javadoc for Business Services, page 7.

Reviewing Input and Response Interfaces

To review information about the classes and fields that are used by these web service operations, you can review the input and response interface tables. Input and response interface tables list the classes and fields used by each operation, the key fields, the data types of each fields, and which fields or classes are required for each action.

See Chapter 2, "Accessing Additional Information about Business Services," page 7 and Appendix A, "Appendix A: Input and Response Interfaces," page 177.

Prerequisites

Before you use the Voucher Match Manager web service, or any of the related operations, you must install and set up the JD Edwards EnterpriseOne Procurement and JD Edwards EnterpriseOne Accounts Payable systems.

See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Setting Up the Procurement System".

See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Setting Up the Accounts Payable System".

processVoucherMatch

This section provides an overview of the processVoucherMatch web service operation, lists prerequisites, and discusses how to set processing options for Purchase Order XPI (P43XPI).

Understanding the processVoucherMatch Web Service Operation

The processVoucherMatch operation is an inbound transaction operation that enables consumers to process voucher match information within the JD Edwards EnterpriseOne system. The consumer can add voucher match records to the JD Edwards EnterpriseOne Procurement system for logged vouchers and both two-way and three-way voucher match. This operation also supports additional charge lines for purchase order detail lines on a voucher.

If the operation completes successfully, the system returns a confirmation message to the consumer. This message contains voucher match data. The system also returns to the consumer any error messages that the operation encounters during processing.

The processVoucherMatch operation processes a voucher match in the JD Edwards EnterpriseOne system. The first call of the XPIInboundVoucherMatch business function (B4302300) checks for record reservation. If the business function finds that any record is reserved, the system stores the error in the Message List and returns from flow.

For an additional charge line, the XPIInboundVoucherMatch business function passes tax information into the taxRateArea, taxExplanationCode, additionalChargeQty, amount, lineTaxableCode, description1, lineTaxAmount, and expenseAcctNumber fields.

The system uses the following processing logic to handle tax for additional charge lines:

1. Pass the following information to the VM Edit Line if the operation passes a non-zero value to the lineTaxAmount field:
    a. Set the lineTaxableCode to *Y*.
    b. Use the taxRateArea and taxExplanationCode if the operation passes these values.
    c. Use the values from the first PO Detail record fetched for the passed in PO header information if either the taxRateArea or the taxExplanationCode value passed in is blank.
    d. Use the values from the supplier purchasing information's default tax values if either the taxRateArea or the taxExplanationCode value passed in is blank.

2. Pass the following information to the VM Edit Line if the operation passes a non-empty and non *N* value for the lineTaxableCode field:
   a. Use the values the operation passes in for the taxRateArea and taxExplanationCode fields.
   b. Use the values from the first PO Detail record fetched for the passed in PO header information if either the taxRateArea or the taxExplanationCode value passed in is blank.
   c. Use the values from the supplier purchasing information's default tax values if either the taxRateArea or the taxExplanationCode value passed in is blank.
3. Do not pass in the tax information to the VM Edit Line function if the integration passes in *N* or blank lineTaxableCode value.

The system uses the expense account number if the B4302300 business function passes it into the system. If the expense account number is not passed in then the system uses the default account number from the P43XPI processing options.

Supported Functionality

This section discusses the functionality that the processVoucherMatch operation supports.

Note. If functionality is not explicitly documented as supported functionality, it is to be understood that the functionality is not supported by the integration solution.

The consumer can add voucher match records to the JD Edwards EnterpriseOne Procurement system. The operation supports both two-way and three-way voucher match. The consumer can also create a logged voucher. The operation processes only one line for the logged voucher.

The processVoucherMatch operation does not support the following functionality:

- Change voucher
- Delete voucher
- Reverse voucher
- Voucher redistribution
- Recost vouchers

Setup Considerations

Before you use this operation, you can set business service properties to define how the system processes voucher match data. If you do not set these business service properties, the system uses default values for each constant. This table includes information about the business service properties used by the processVoucherMatch operation:

Group	Key	Description	Default Value
J43B0010	J43B0010_P43XPI_VERSION	Use this business service property to specify which version of the Purchase Order XPI program (P43XPI) the operation uses.	*ZJDE0001*

Group	Key	Description	Default Value
J43B0010	J43B0010_BYPASS_ BSFN_WARNINGS	Use this business service property to specify whether the system converts warning messages to errors during processing.	*1* **Note.** The system does not treat warnings as errors so the system keeps processing.
J43B0010	J43B0010_PREFIX_1	Use this business service property to specify the prefix value the operation uses for error messages when JD Edwards EnterpriseOne processes detail lines.	*VoucherMatchRecordSentin*
J43B0010	J43B0010_PREFIX_2	Use this business service property to specify the prefix value the operation uses for error messages when JD Edwards EnterpriseOne processes additional charges.	*VoucherMatchAdditional RecordSentin*

The processVoucherMatch web service operation calls the XPIInboundVoucherMatch business function (B4302300) which calls the Purchase Order XPI program (P43XPI). You must set the processing options for the P43XPI program, which provides the input for the voucher type. You must set the Voucher Type processing option on the Voucher Match tab.

See Chapter 18, "Voucher Match Manager Web Service," Prerequisites, page 173.

Implementation Details

The following table includes information that can help determine whether the processVoucherMatch operation is functioning correctly:

Question	Answer
How can I tell if the operation completes successfully?	If a business service exception is not thrown, then the operation completed successfully and the system returns a confirmation message to the consumer. The return message includes all of the fields that are listed in the response interface for this operation. However, some of those fields can contain blank or zero values, depending on the data that exists in the JD Edwards EnterpriseOne system. At a minimum, the system returns non-zero values for these fields: • supplierInvoiceNumber • documentNumber • documentTypeCode • documentCompany • documentCompany • messages • documentLineNumber • documentLineNumberSuffix
If I encounter errors while processing a transaction, do I need to reverse the transaction?	This operation uses standard transaction processing. Therefore, if you encounter errors during processing, the system does not update any information in the JD Edwards EnterpriseOne system. No manual update is necessary.
Does this operation use record reservation?	This operation calls the XPIInboundVoucherMatch business function (B4302300). This function reserves records in the JD Edwards EnterpriseOne system when the operation adds voucher match records for both two-way and three-way voucher match.

Prerequisites

Before using the processVoucherMatch operation, you must:

- Set the Voucher Type processing option on the Voucher Match tab of the P43XPI program.

  Note. It is recommended that you set the Voucher Type processing option to *1*. This processing option setting enables the operation to process matched and logged vouchers.

- Set the 2. Voucher Match Processing Option Version (P4314) processing option on the Versions tab of the P43XPI program.
- Set the 2. Expense Account for Additional Charges processing option on the Voucher Match tab of the P43XPI program for additional charges.
- Set the processing options for the version of P4314 called by the P43XPI program.

  See *JD Edwards EnterpriseOne Procurement Management 9.0 Implementation Guide*, "Creating Vouchers," Setting Processing Options for Voucher Match (P4314).

- Set the 8. Orders Per Voucher processing option on the Process tab of the P4314 program to *1*.
- Set the 1. Order Line Entry processing option on the New Order Line tab of the P4314 program to *2* for additional charges.
- Set the 2. Line Type processing option on the New Order Line tab of the P4314 program to *J* for additional charges.
- Set the 3. Last Status Code processing option on the New Order Line tab of the P4314 program to *400* for additional charges.
- Set the 3. AP Master Business Function Processing Option Version for Logged Vouchers (P0400047) processing option on the Versions tab of the P43XI program.
- Set the processing options for the version of P0400047 called by the P43XPI program.

  See *JD Edwards EnterpriseOne Accounts Payable 9.0 Implementation Guide*, "Processing Accounts Payable Vouchers," Setting Processing Options for Voucher Entry MBF (P0400047).

Note. If logged vouchers are enabled in the Voucher Type processing option on the Voucher Match tab of the P43XPI program, then you must set the 1. Voucher Logging processing option on the Logging tab of the P0400047 program.

Setting Processing Options for Purchase Order XPI (P43XPI)

Processing options enable you to specify the default processing for programs and reports.

For programs, you can specify options such as the default values for specific transactions, whether fields appear on a form, and the version of the program that you want to run.

Order Statuses

1. Beginning Order Status allowed for External Publish	Specify a value from UDC 40/AT that indicates the last step in the processing cycle that this order line has successfully completed.
2. Ending Order Status allowed for External Publish	Specify a value from UDC 40/AT that indicates the next step in the order flow of the line type.
3. Acknowledged Order Status Code	Specify a value from UDC 40/AT that indicates the next step in the order flow of the line type.
4. Acknowledged With Change Order Status Code	Specify a value from UDC 40/AT that indicates the next step in the order flow of the line type.
5. Pending Order Status Code	Specify a value from UDC 40/AT indicates the next step in the order flow of the line type.

Versions

1. Purchase Order Entry Processing Option Version (P4310)	Specify the version that the system uses when you enter a purchase order.

2. Voucher Match Processing Option Version (P4314)

Specify the version that the system uses when you match an invoice to a purchase order line or receipt line. If the Voucher Match processing option, which is located on the Voucher Match tab, is set to 1 or 2, the system validates the version that you specify.

3. AP Master Business Function Processing Option Version for Logged Vouchers (P0400047)

Specify the version that the system uses when you log a voucher for an invoice. If the Voucher Match processing option, which is located on the Voucher Match tab, is set to blank or 1, the system validates the version that you specify to ensure that the voucher logging process is active.

Receipts

1. Shipped Order Status Code

Specify a value from UDC 40/AT that indicates the next step in the order flow of the line type.

2. Advance Status Only

Specify whether the system advances the shipment status of the line or order. Values are:

Blank: Do not advance to Shipped Status.

1: Advance to shipped status (no receipt).

Voucher Match

1. Voucher Type

Specify whether the system creates a logged voucher or matched voucher. Values are:

Blank: The system always creates a logged voucher.

1: If the invoice includes specific purchase order information, the system creates a matched voucher. If the invoice does not include specific purchase order information, the system creates a logged voucher.

2: The system always creates a matched voucher.

2. Expense Account for Additional Charges

Specify the account that the system uses to expense the additional charges that are specified on the invoice. You set up the account format in the General Accounting Constants program (P0000), using one of the following formats for account numbers:

- Structured account (business unit.object.subsidiary)
- 25-digit unstructured number
- 25-digit unstructured number
- 8-digit short account ID number
- Speed code

APPENDIX A

Appendix A: Input and Response Interfaces

This appendix lists the input and response interface data for all business services that are included in this reference guide.

The input interface tables also specify which fields or classes are required when performing a specific action. If a field or class is required, a Y is entered in the table under the specified action column for that field or class. These actions are included in the table:

- Add (Column heading A in the tables.)
- Change (Column heading C in the tables.)
- Delete or Cancel (Column heading D/C in the tables.)
- Inquire (Column heading I in the tables.)

Note. If a class is specified as a required class, typically, you can use one or more of the fields included in that class to satisfy the requirement. For example, if an Entity class is marked as a required class, you can typically use either the entityId, entityLongId, or entityTaxId field to satisfy the requirement.

See Also

Chapter 2, "Accessing Additional Information about Business Services," Reviewing Input and Response Interfaces, page 29

Accounts Payable

This section lists these interface tables:

- getVoucher: Input Interface
- getVoucher: Response Interface
- processVoucher: Input Interface
- processVoucher: Response Interface

getVoucher: Input Interface

This table lists the input interface information for the getVoucher web service operation:

GetVoucher - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
VoucherGetHeader							
	businessUnit	String					
	company	String					
	supplierInvoiceNumber	String					
	dateInvoice	Calendar					
	dateAccounting	Calendar					
	remark	String					
VoucherKey							
	documentNumber	Integer					Y
	documentTypeCode	String					Y
	documentCompany	String					Y
Entity - supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
VoucherGetDetail							
VoucherLineKey							
	documentPayItem	String					Y
	documentPayItemExtension	Integer					Y

getVoucher – Input Interface

getVoucher: Response Interface

The getVoucher web service operation returns the ShowVoucher message as the response interface. These tables list the response interface information for the getVoucher web service operation:

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
VoucherShowHeader[]		
	company	String
	businessUnit	String
	supplierInvoiceNumber	String
	rateExchange	Integer
	currencyCode	String
	currencyModeCode	String
	batchNumber	Integer
	batchTypeCode	String
	dateBatch	Calendar
VoucherKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
VoucherShowDetail		
VoucherResult[]		
	documentTypeAdjustingCode	String
Entity - supplier		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	entityIdPayee	Integer
	entityIdApprover	Integer
	dateInvoice	Calendar
	dateServiceTax	Calendar
	dateDueNet	Calendar
	dateDueDiscount	Calendar
	dateAccounting	Calendar
	fiscalYear	Integer
	century	Integer
	periodNumber	Integer
	balancedJournalEntries	String

getVoucher – Response Interface (1 of 7)

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
	statusCodePay	String
	amountGrossDomestic	BigDecimal
	amountOpenDomestic	BigDecimal
	amountDiscountAvailableDomestic	BigDecimal
	amountDiscountTakenDomestic	BigDecimal
	amountTaxableDomestic	BigDecimal
	amountNonTaxableDomestic	BigDecimal
	amountTaxDomestic	BigDecimal
	taxRateAreaCode	String
	taxExplanationCode	String
	amountGrossForeign	BigDecimal
	amountOpenForeign	BigDecimal
	amountDiscountAvailableForeign	BigDecimal
	amountDiscountTakenForeign	BigDecimal
	amountTaxableForeign	BigDecimal
	amountNonTaxableForeign	BigDecimal
	amountTaxForeign	BigDecimal
	glOffsetCode	String
	bankAccountNumber	String
	glPostedCode	String
	bankTransitId	String
	paymentTermsCode	String
	voidFlag	String
	sequenceNumber	BigDecimal
	reference	String
	unitNumber	String
	businessUnit2	String
	remark	String
	frequencyRecurring	String
	frequencyRecurringNumberOfPayments	Integer
	checksControl	String
	paymentFinal	String

getVoucher – Response Interface (2 of 7)

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
	numberOfUnits	BigDecimal
	unitOfMeasure	String
	paymentInstrumentCode	String
	taxRateAreaWithholdingCode	String
	taxExplanationWithholdingCode	String
	miscellaneousCodeArAp1	String
	miscellaneousCodeArAp2	String
	miscellaneousCodeArAp3	String
	flag1099	String
	multiCurrencyDistribution	String
	rateExchangeHistorical	Integer
	dateHistorical	Calendar
	transactionOriginator	String
	bankTransitNumber	String
	paymentHandlingCode	String
	taxStatusDeferred	String
	currencyCodeBase	String
	amountDistributeToDomestic	BigDecimal
	amountDistributeToForeign	BigDecimal
	amountTaxNonRecoverableDomestic	BigDecimal
	amountTaxNonRecoverableForeign	BigDecimal
	isPurchasingOriginated	Boolean
	isSummarized	Boolean
	payWhenPaid	String
	payWhenPaidGroupNumber	Integer
	nettingTransaction	Integer
	nettingDocumentNumber	Integer
	nettingStatusCode	String
VoucherLineKey		
	documentPayItem	String
	documentPayItemExtension	Integer

getVoucher – Response Interface (3 of 7)

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
OriginalOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	documentSuffix	String
PurchaseOrderLineKey		
	documentLineNumber	BigDecimal
PurchaseOrderKey		
	documentNumber	String
	documentTypeCode	String
	documentCompany	String
	documentSuffix	String
CategoryCodesAccountsPayable		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
UserReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
GeneralLedger[]		
	glPostedCode	String
	batchNumber	Integer

getVoucher – Response Interface (4 of 7)

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
	batchTypeCode	String
	dateBatch	Calendar
	dateBatchSystem	Calendar
	timeBatch	Integer
	company	String
	accountId	String
	accountLongId	String
	accountMode	String
	periodNumber	Integer
	century	Integer
	fiscalYear	Integer
	currencyCodeFrom	String
	rateExchange	Integer
	rateExchangeHistorical	Integer
	dateHistorical	Calendar
	amountDomestic	BigDecimal
	amountForeign	BigDecimal
	numberOfUnits	BigDecimal
	unitOfMeasure	String
	glOffsetCode	String
	isReverseOrVoid	Boolean
	nameExplanationAlpha	String
	nameExplanationRemark	String
	reference1	String
	reference2	String
	reference3	String
	checkNumber	String
	dateCheck	Calendar
	dateCheckCleared	Calendar
	serialNumber	String
	postCodeBatchRearEnd	String
	reconciledCode	String

getVoucher – Response Interface (5 of 7)

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
	isSummarized	Boolean
	purgeCode	String
	flag1099	String
	deleteNotAllowed	String
	clientFreeForm1	String
	clientFreeForm2	String
	postCodeLeaseCostLedger	String
	billCode	String
	dateInvoice	Calendar
	categoryWorkOrderCode	String
	fiscalYearWeekly	Integer
	fiscalPeriodWeekly	Integer
	paymentFinal	String
	sequenceNumber	BigDecimal
	jobCategory	String
	jobStep	String
	businessUnitHome	String
	divisionOfInterest	Integer
	leaseNumberAlternate	String
	leaseTypeAlternateCode	String
	dateServiceTax	Calendar
	transactionOriginator	String
	registrationNumber	Integer
	paymentId	Integer
	currencyCodeBase	String
	taxItemNumber	Integer
	activityBasedCostingCode	String
	distributionLineNumber	BigDecimal
	receiptNumber	String
GeneralLedgerKey		
	documentNumber	Integer
	documentTypeCode	String

getVoucher – Response Interface (6 of 7)

ShowVoucher - Response Interface		
Class	**Field**	**Data Type**
	documentCompany	String
	documentLineNumber	Integer
	dateAccounting	Calendar
	lineExtensionCode	String
	ledgerTypeCode	String
GLAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
Subledger		
	subledger	String
	subledgerTypeCode	String
GLPostCodes		
	postCodeAlternate0	String
	postCodeAlternate1	String
	postCodeAlternate2	String
	postCodeAlternate3	String
	postCodeAlternate4	String
	postCodeAlternate5	String
	postCodeAlternate6	String
	postCodeAlternate7	String
	postCodeAlternate8	String
	postCodeAlternate9	String
GLCostObjects		
	costObjectCode1	String
	costObjectCode2	String
	costObjectCode3	String
	costObjectCode4	String
	costObjectTypeCode1	String
	costObjectTypeCode2	String
	costObjectTypeCode3	String
	costObjectTypeCode4	String

getVoucher – Response Interface (7 of 7)

processVoucher: Input Interface

These tables lists the input interface information for the processVoucher web service operation:

ProcessVoucher - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
ProcessVoucherHeader							
	businessUnit	String					
	company	String	Y				
	paymentTermsCode	String					
	paymentInstrumentCode	String					
	supplierInvoiceNumber	String					
	dateInvoice	Calendar					
	dateAccounting	Calendar	Y				
	dateServiceTax	Calendar					
	dateBatch	Calendar					
	currencyCode	String					
	rateExchange	Integer					
	currencyModeCode	String					
	remark	String					
ProcessVoucherHeader.processing							
	actionType	String	Y		Y		
	processingVersion	String					
	processingVersionGeneralLedger	String					
ProcessVoucherHeader.VoucherKey							
	documentNumber	String			Y		
	documentTypeCode	String			Y		
	documentCompany	String			Y		
ProcessVoucherHeader.Entity - supplier							
	entityId	Integer	Y				
	entityLongId	String	Y				
	entityTaxId	String					

processVoucher – Input Interface (1 of 4)

ProcessVoucher - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
ProcessVoucherHeader.Entity - approver							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
ProcessVoucherHeader.PurchaseOrderKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
ProcessVoucherHeader.ProcessVoucherDetail							
ProcessVoucherHeader.ProcessVoucherDetail.Voucher[]							
	paymentTermsCode	String					
	remark	String					
	glOffsetCode	String					
	amountGrossDomestic	BigDecimal	Y				
	amountGrossForeign	BigDecimal					
	amountDiscountAvailableDomestic	BigDecimal					
	amountDiscountAvailableForeign	BigDecimal					
	amountDiscountTakenDomestic	BigDecimal					
	amountDiscountTakenForeign	BigDecimal					
	amountTaxableDomestic	BigDecimal					
	amountTaxableForeign	BigDecimal					
	amountNonTaxableDomestic	BigDecimal					
	amountNonTaxableForeign	BigDecimal					
	amountTaxDomestic	BigDecimal					
	amountTaxForeign	BigDecimal					
	amountTaxNonRecoverableDomestic	BigDecimal					
	amountTaxNonRecoverableForeign	BigDecimal					
	dateDue	Calendar					
	statusCodePay	String					
	taxRateAreaCode	String					

processVoucher – Input Interface (2 of 4)

ProcessVoucher - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	taxExplanationCode	String					
	frequencyRecurring	String					
	frequencyRecurringNumberOfPayments	String					
ProcessVoucherHeader.ProcessVoucherDetail.Voucher[].VoucherLineKey							
	documentPayItem	String					
	documentPayItemExtension	Integer					
ProcessVoucherHeader.ProcessVoucherDetail.Voucher[].Entity - Payee							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
ProcessVoucherHeader.ProcessVoucherDetail.Voucher[]. PurchaseOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
ProcessVoucherHeader.ProcessVoucherDetail.Voucher[].UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					
ProcessVoucherHeader.ProcessVoucherDetail.GLDistribution[]							
	documentLineNumber	Integer					
	amountDomestic	BigDecimal	Y				
	amountForeign	BigDecimal					
	nameExplanationRemark	String	Y				
	reference	String					
	assetId	String					
ProcessVoucherHeader.ProcessVoucherDetail.GLDistribution[].Subledger							
	subledger	Integer					
	subledgerTypeCode	String					

processVoucher – Input Interface (3 of 4)

ProcessVoucher - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
ProcessVoucherHeader.ProcessVoucherDetail.GLDistribution[].GLAccount							
	businessUnit	String	Y				
	objectAccount	String	Y				
	subsidiary	String					
ProcessVoucherHeader.ProcessVoucherDetail.GLDistribution[].GLAccountKey							
	accountId	String	Y				
	accountLongId	String					
	accountAlternate	String					
ProcessVoucherHeader.ProcessVoucherDetail.GLDistribution[].GLPostCodes							
	postCodeAlternate0	String					
	postCodeAlternate1	String					
	postCodeAlternate2	String					
	postCodeAlternate3	String					
	postCodeAlternate4	String					
	postCodeAlternate5	String					
	postCodeAlternate6	String					
	postCodeAlternate7	String					
	postCodeAlternate8	String					
	postCodeAlternate9	String					
ProcessVoucherHeader.ProcessVoucherDetail.GLDistribution[].GLCostObjects							
	costObjectCode1	String					
	costObjectCode2	String					
	costObjectCode3	String					
	costObjectCode4	String					
	costObjectTypeCode1	String					
	costObjectTypeCode2	String					
	costObjectTypeCode3	String					
	costObjectTypeCode4	String					

processVoucher – Input Interface (4 of 4)

processVoucher: Response Interface

The procesVoucher web service operation returns the ConfirmProcessVoucher message as the response interface. This table lists the response interface information for the processVoucher web service operation:

ConfirmProcessVoucher - Response Interface		
Class	**Field**	**Data Type**
ConfirmVoucherHeader		
VoucherKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
ConfirmVoucherDetail[]		
VoucherLineKey[]		
	documentPayItem	String
	documentPayItemExtension	Integer
GeneralLedgerKey[]		
	documentLineNumber	Integer
	lineExtensionCode	String
messages[]		
	messages	String

processVoucher – Response Interface

Address Book

This section lists these interface tables:

- getAddressBook: Input Interface
- getAddressBook: Response Interface
- processAddressBook: Input Interface
- processAddressBook: Response Interface
- getContact: Input Interface
- getContact: Response Interface
- processContact: Input Interface
- processContact: Response Interface

getAddressBook: Input Interface

These tables list the input interface information for the getAddressBook web service operation:

getAddressBook - Input Interface			Required				Key
Class	**Field**	**Data Type**					**(Y/N)**
	entityName	String					
	entityTypeCode	String					
	businessUnit	String					
	industryClassificationCode	String					
	languageCode	String					
Entity							
	entityId	Integer					Y
	entityLongId	String					
	entityTaxId	String					
GetAddress							
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
CategoryCodesAddressBook							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					

getAddressBook – Input Interface (1 of 2)

getAddressBook - Input Interface			Required				Key
Class	Field	Data Type					(Y/N)
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					
	categoryCode024	String					
	categoryCode025	String					
	categoryCode026	String					
	categoryCode027	String					
	categoryCode028	String					
	categoryCode029	String					
	categoryCode030	String					

getAddressBook – Input Interface (2 of 2)

getAddressBook: Response Interface

The getAddressBook web service operation returns the ShowAddressBook message as the response interface. These tables list the response interface information for the getAddressBook web service operation:

ShowAddressBook - Response Interface		
Class	**Field**	**Data Type**
AddressBookResult[]		
	entityTypeCode	String
	isEntityTypeNettingIndicator	Boolean
	isEntityTypeEmployee	Boolean
	isEntityTypePurchaser	Boolean
	isEntityTypeReceivables	Boolean
	isEntityTypePayables	Boolean
	isEntityType3	Boolean
	isEntityType4	Boolean
	isEntityType5	Boolean
	entityName	String
	entityNameSecondary	String
	entityTaxIdAdditional	String
	businessUnit	String
	languageCode	String
	industryClassificationCode	String
	remark	String
	description1	String
	creditMessageCode	String
	personCorporationCode	String
	subledgerInactiveCode	String
	taxExemptCertificate	String
	employeeGroupApprovalsCode	String
Entity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
RelatedAddress		
	entityIdRelated1	Integer
	entityIdRelated2	Integer
	entityIdRelated3	Integer
	entityIdRelated4	Integer
	entityIdRelated5	Integer

getAddressBook – Response Interface (1 of 4)

ShowAddressBook - Response Interface		
Class	**Field**	**Data Type**
	entityIdRelated6	Integer
Parent[]		
	entityIdParent	Integer
	parentStructureTypeCode	String
Address		
	mailingName	String
	mailingNameSecondary	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
	dateEffective	Calendar
CategoryCodesAddressBook		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String

getAddressBook – Response Interface (2 of 4)

ShowAddressBook - Response Interface		
Class	**Field**	**Data Type**
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
	categoryCode021	String
	categoryCode022	String
	categoryCode023	String
	categoryCode024	String
	categoryCode025	String
	categoryCode026	String
	categoryCode027	String
	categoryCode028	String
	categoryCode029	String
	categoryCode030	String
Classifications		
	classificationCode001	String
	classificationCode002	String
	classificationCode003	String
	classificationCode004	String
	classificationCode005	String
UserReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
ShowStatistics		
	numberOfEmployees	Integer
	rateGrowth	Integer
	revenueRangeCode	String
	glBankAccount	String
	yearCompanyFounded	String

getAddressBook – Response Interface (3 of 4)

ShowAddressBook - Response Interface		
Class	**Field**	**Data Type**
	dunBradStreetId	String
Stock		
	stockTickerSymbol	String
	stockExchange	String
PhoneNumber[]		
	contactId	Integer
	relatedPersonId	String
	phoneLineNumber	String
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddress[]		
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	Integer
Messages[]		

getAddressBook – Response Interface (4 of 4)

processAddressBook: Input Interface

These tables list the input interface information for the processAddressBook web service operation:

ProcessAddressBook - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
AddressBook							
	entityTypeCode	String	Y				
	businessUnit	String					
	entityName	String	Y				
	entityNameSecondary	String					
	languageCode	String					
	industryClassificationCode	String					
	remark	String					
	callSalesTeamAlignment	String					
	creditMessageCode	String					
	personCorporationCode	String					
	entityTaxIdAdditional	String					
	taxExemptCertificate	String					
	isEntityTypeNettingIndicator	Boolean					
	isEntityTypePayables	Boolean					
	isEntityTypeReceivables	Boolean					
	isEntityTypePurchaser	Boolean					
	isEntityTypeEmployee	Boolean					
	employeeGroupApprovalsCode	String					
processing							
	actionType	String	Y	Y	Y	Y	
	processingVersion	String					
Entity - entity							
	entityId	Integer		Y	Y	Y	Y
	entityLongId	String					
	entityTaxId	String					
Entity - parent							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

processAddressBook – Input Interface (1 of 5)

ProcessAddressBook - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Entity - relatedAddress1							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - relatedAddress2							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - relatedAddress3							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - relatedAddress4							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - relatedAddress5							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - relatedAddress6							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
CategoryCodesAddressBook							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					

processAddressBook – Input Interface (2 of 5)

ProcessAddressBook - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					
	categoryCode024	String					
	categoryCode025	String					
	categoryCode026	String					
	categoryCode027	String					
	categoryCode028	String					
	categoryCode029	String					
	categoryCode030	String					
Classifications							
	classificationCode001	String					
	classificationCode002	String					
	classificationCode003	String					
	classificationCode004	String					

processAddressBook – Input Interface (3 of 5)

ProcessAddressBook - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	classificationCode005	String					
Address							
	mailingName	String					
	mailingNameSecondary	String					
	addressLine1	String					
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
	dateEffective	Calendar					
Stock							
	stockTickerSymbol	String					
	stockExchange	String					
Statistics							
	numberOfEmployees	Integer					
	rateGrowth	Integer					
	revenueRangeCode	String					
	yearCompanyFounded	String					
	dunBradStreetId	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					

processAddressBook – Input Interface (4 of 5)

ProcessAddressBook - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
PhoneNumber[]							
	actionType	String	Y	Y	Y	Y	
	contactId	Integer					Y
	phoneLineNumber	Integer		Y	Y	Y	Y
	areaCode	String					
	phoneTypeCode	String					
	phoneNumber	String					
ElectronicAddress[]							
	actionType	String	Y	Y	Y	Y	Y
	contactId	Integer					Y
	electronicAddressLineNumber	Integer		Y	Y	Y	Y
	electronicAddressTypeCode	String					
	electronicAddress	String					
	electronicAddressClassificationCode	String					
	messageIndicatorCode	Integer					

processAddressBook – Input Interface (5 of 5)

processAddressBook: Response Interface

These tables list the response interface information for the processAddressBook web service operation:

ConfirmAddressBook - Response Interface		
Class	**Field**	**Data Type**
	entityTypeCode	String
	businessUnit	String
	entityName	String
	languageCode	String
Entity		
	EntityId	Integer
	EntityLongId	String
	EntityTaxId	String
Address		
	mailingName	String
	mailingNameSecondary	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
	dateEffective	Date
PhoneNumber[]		
	actionType	String
	contactId	Integer
	phoneLineNumber	Integer
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddress[]		
	actionType	String
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String

processAddressBook – Response Interface (1 of 2)

ConfirmAddressBook - Response Interface		
Class	**Field**	**Data Type**
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	Integer
messages[]		
	messages	String

processAddressBook – Response Interface (2 of 2)

getContact: Input Interface

These tables list the input interface information for the getContact web service operation:

getContact - Input Interface			Required				Key
Class	**Field**	**Data Type***	**A**	**C**	**D/C**	**I**	**(Y/N)**
	contactId	Integer					Y
	entityNameContact	String					
	mailingName	String					
	nickName	String					
	contactTitle	String					
	remark	String					
	salutationName	String					
	givenName	String					
	middleName	String					
	surname	String					
	typeCode	String					
	description1	String					
	displaySequence	String					
	contactTypeCode	String					
	functionCode	String					
	preferredContactMethodCode	String					
	primaryContactCode	String					
	dayOfBirth	Integer					
	monthOfBirth	Integer					
	yearOfBirth	Integer					
	genderCode	String					
	mailingNameSecondary	String					
	entityNameSecondary	String					
entity							
	entityId	Integer					Y
	entityLongId	String					
	entityTaxId	String					
categoryCodesContact							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					

getContact – Input Interface (1 of 2)

getContact - Input Interface			Required				Key
Class	Field	Data Type*	A	C	D/C	I	(Y/N)
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					

getContact – Input Interface (2 of 2)

getContact: Response Interface

The getContact web service operation uses the ShowContact message as the response. These tables list the response interface information for the getContact web service operation:

ShowContact - Response Interface		
Class	**Field**	**Data Type**
ContactRecord[]		
	contactId	Integer
	entityNameContact	String
	mailingName	String
	nickName	String
	contactTitle	String
	remark	String
	salutationName	String
	givenName	String
	middleName	String
	surname	String
	typeCode	String
	description1	String
	displaySequence	String
	contactTypeCode	String
	functionCode	String
	preferredContactMethodCode	String
	primaryContactCode	String
	dayOfBirth	Integer
	monthOfBirth	Integer
	yearOfBirth	Integer
	genderCode	String
	mailingNameSecondary	String
	entityNameSecondary	String
entity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
categoryCodesContact		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String

getContact – Response Interface (1 of 3)

ShowContact - Response Interface		
Class	**Field**	**Data Type**
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
AlternateAddressRecord[]		
	dateEffective	Date
	contactAddressTypeCode	String
	mainAddressFlagCode	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
PhoneNumberRecord[]		
	contactId	Integer
	relatedPersonId	String

getContact – Response Interface (2 of 3)

ShowContact - Response Interface		
Class	**Field**	**Data Type**
	phoneLineNumber	String
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddressRecord[]		
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	Integer
Messages[]		

getContact – Response Interface (3 of 3)

processContact: Input Interface

These tables list the input interface information for the processContact web service operation:

processContact - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
ProcessContact (ProcessContact)							
	contactId	Integer		Y	Y		Y
	entityNameContact	String	Y	Y			
	mailingName	String	Y	Y			
	nickName	String					
	contactTitle	String					
	remark	String					
	salutationName	String					
	givenName	String					
	middleName	String					
	surname	String					
	typeCode	String					
	description1	String					
	displaySequence	BigDecimal					
	contactTypeCode	String					
	functionCode	String					
	preferredContactMethodCode	String					
	primaryContactCode	String					
	dayOfBirth	Integer					
	monthOfBirth	Integer					
	yearOfBirth	Integer					
	genderCode	String					
	mailingNameSecondary	String					
	entityNameSecondary	String					
Processing							
	actionType	String	Y	Y	Y		
	processingVersion	String					
Entity							
	entityId	Integer	Y	Y	Y		Y
	entityTaxId	String					
	entityLongId	String					

processContact – Input Interface (1 of 3)

processContact - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
CategoryCodesContact							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
AlternateAddress[]							
	actionType	String	Y	Y	Y		
	dateEffective	Data	Y	Y	Y		Y
	contactAddressTypeCode	String	Y	Y	Y		Y
	mainAddressFlagCode	String					
	addressLine1	String	Y				
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					

processContact – Input Interface (2 of 3)

processContact - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
PhoneNumber[]							
	actionType	String	Y	Y	Y		
	contactId	Integer	Y	Y	Y		Y
	phoneLineNumber	Integer		Y	Y		Y
	areaCode	String					
	phoneTypeCode	String					
	phoneNumber	String	Y				
ElectronicAddress[]							
	actionType	String	Y	Y	Y		Y
	contactId	Integer	Y	Y	Y		Y
	electronicAddressLineNumber	Integer		Y	Y		Y
	electronicAddressTypeCode	String	Y	Y	Y		
	electronicAddress	String	Y				
	electronicAddressClassificationCode	String					
	messageIndicatorCode	String					

processContact – Input Interface (3 of 3)

processContact: Response Interface

The processContact web service operation uses the ConfirmProcessContact message as the response. These tables list the response interface information for the processContact web service operation:

ConfirmProcessContact - Response Interface		
Class	**Field**	**Data Type**
ProcessContact		
	contactId	Integer
	entityNameContact	String
	mailingName	String
	nickName	String
	contactTitle	String
	remark	String
	salutationName	String
	givenName	String
	middleName	String
	surname	String
	typeCode	String
	description1	String
	displaySequence	BigDecimal
	contactTypeCode	String
	functionCode	String
	preferredContactMethodCode	String
	primaryContactCode	String
	dayOfBirth	Integer
	monthOfBirth	Integer
	yearOfBirth	Integer
	genderCode	String
	mailingNameSecondary	String
	entityNameSecondary	String
Entity		
	entityId	Integer
	entityTaxId	String
	entityLongId	String
AlternateAddress[]		
	actionType	String
	dateEffective	Data
	contactAddressTypeCode	String
	mainAddressFlagCode	String

processContact – Response Interface (1 of 2)

ConfirmProcessContact - Response Interface		
Class	**Field**	**Data Type**
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
PhoneNumber[]		
	actionType	String
	contactId	Integer
	phoneLineNumber	Integer
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddress[]		
	actionType	String
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	String
Messages[]		
	messages	String

processContact – Response Interface (2 of 2)

Capital Assets

This section lists these interface tables:

- createCapitalAssetConditionBasedAlert: Input Interface
- createCapitalAssetConditionBasedAlert: Response Interface
- getCapitalAssetConditionBasedAlert: Input Interface
- getCapitalAssetConditionBasedAlert: Response Interface

createCapitalAssetConditionBasedAlert: Input Interface

These tables list the input interface information for the createCapitalAssetConditionBasedAlert web service operation:

createCapitalAssetConditionBasedAlert - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
Processing							
	processingVersion	String					
Alert							
	description	String	Y				
	problemDescription	String					
	alertLevelCode	String	Y				
	alertStatusCode	String					
	sendNotificationMessage	Boolean	Y				
	notificationStructureTypeCode	String					
	automatedResponseTypeCode	String					
	dateEstimatedStart	Calendar					
	investigationStructureTypeCode	String					
	modelWorkOrder	Integer					
	serviceTypeCode	String					
	dateRequested	Calendar	Y				
	timeRequested	Integer	Y				
	eventTimeUTC	Integer					
	timeZoneCode	String					
	daylightSavingsRuleCode	String					
Equipment							
	itemAsset	Integer	Y				
	unitNumber	String					
	serialNumber	String					
	productModelCode	String					
	measurementLocationCode	String					
Site							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

createCapitalAssetConditionBasedAlert – Input Interface (1 of 3)

createCapitalAssetConditionBasedAlert - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
NotificationRecipient							
	entityId	Integer	Y				
	entityLongId	String					
	entityTaxId	String					
InvestigationRecipient							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Address							
	reference1	String					
	reference2	String					
	mailingName	String					
	addressLine1	String					
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					
	countyCode	String					
	stateCode	String					
	zipPostalCode	String					
	countryCode	String					
PhoneNumber							
	areaCode	String					
	phoneNumber	String					
	phoneTypeCode	String					
	phoneLineNumber	String					

createCapitalAssetConditionBasedAlert – Input Interface (2 of 3)

createCapitalAssetConditionBasedAlert - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	contactId	Integer					
	contactPersonalId	Integer					
UserReservedData							
	userReservedAmount	BigDecimal					
	userReservedCode	String					
	userReservedReference	String					
UserReservedChar							
	userReservedChar1	String					
	userReservedChar2	String					
UserReservedDate							
	userReservedDate	Calendar					
	userReservedDate1	Calendar					
	userReservedDate2	Calendar					
UserReservedNumeric							
	userReservedNumeric1	BigDecimal					
	userReservedNumeric2	BigDecimal					
UserReservedNumber							
	userReservedNumber	Integer					
	userReservedNumber1	Integer					
	userReservedNumber2	Integer					
UserReservedString							
	userReservedString1	String					
	userReservedString2	String					
	userReservedString3	String					
	userReservedString4	String					
	userReservedString5	String					
	userReservedString6	String					
	userReservedString7	String					
	userReservedString8	String					
	userReservedString9	String					
	userReservedString10	String					

createCapitalAssetConditionBasedAlert – Input Interface (3 of 3)

createCapitalAssetConditionBasedAlert: Response Interface

The createCapitalAssetConditionBasedAlert web service operation uses the ConfirmCapitalAssetConditionBasedAlert message as the response. This table lists the response interface information for the createCapitalAssetConditionBasedAlert web service operation:

ConfirmCapitalAssetConditionBasedAlert		
Class	**Field**	**Data Type**
	serialNumber	String
	MeasurementLocation	String
Alert		
	EventDate	Date
	EventTime	Integer
	TimeZone	String
	DaylightSavings	String
	AlertDescription	String
	ProblemDescription	String
	AlertLevel	String
	AlertStatus	String
Response		
	AutomatedResponseType	String
	NotificationStructureType	String
NotificationRecipient		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
messages[]		
	messages	E1Message List

createCapitalAssetConditionBasedAlert – Response Interface

getCapitalAssetConditionBasedAlert: Input Interface

These tables list the input interface information for the getCapitalAssetConditionBasedAlert web service operation:

GetCapitalAssetConditionBasedAlert - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
GetCapitalAssetConditionBasedAlert							
Alert							
	alertId	Integer					
	description1	String					
	problemDescription	String					
	alertLevelCode	String					
	alertStatusCode	String					
	dateEstimatedStart	Calendar					
	modelWorkOrder	Integer					
	serviceTypeCode	String					
	dateRequested	Calendar					
Equipment							
	itemAsset	Integer					
	unitNumber	String					
	serialNumber	Integer					
	measurementLocationCode	String					
	itemTestId	String					
Site							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Originator							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

getCapitalAssetConditionBasedAlert – Input Interface (1 of 2)

GetCapitalAssetConditionBasedAlert - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
NotificationRecipient							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
InvestigationRecipient							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

getCapitalAssetConditionBasedAlert – Input Interface (2 of 2)

getCapitalAssetConditionBasedAlert: Response Interface

The getCapitalAssetConditionBasedAlert web service operation uses the ShowCapitalAssetConditionBasedAlert message as the response. These tables list the response interface information for the getCapitalAssetConditionBasedAlert web service operation:

ShowCapitalAssetConditionBasedAlert - Response Interface		
Class	**Field**	**Data Type**
	numberRowsReturned	long
CapitalAssetConditionBasedAlert[]		
	businessUnit	String
	entityIdSite	Integer
	entityIdCustomer	Integer
	entityIdOriginator	Integer
	entityIdNotificationRecipient	Integer
	entityIdInvestigationRecipient	Integer
Alert		
	alertId	Integer
	alertDescription	String
	problemDescription	String
	alertLevelCode	String
	alertStatusCode	String
	sendNotificationMessage	Boolean
	notificationStructureTypeCode	String
	automatedResponseTypeCode	String
	dateEstimatedStart	Calendar
	investigationStructureTypeCode	String
	investigationMessageSent	Boolean
	investigationCompleted	Boolean
	modelWorkOrder	Integer
	maintenanceServiceType	String
	dateRequested	Calendar
	timeRequested	Integer
	eventTimeUTC	Integer
	timeZoneCode	String
	daylightSavingsRuleCode	String
Equipment		
	itemAsset	Integer
	unitNumber	String
	serialNumber	Integer
	measurementLocationCode	String

getCapitalAssetConditionBasedAlert – Response Interface (1 of 3)

ShowCapitalAssetConditionBasedAlert - Response Interface		
Class	**Field**	**Data Type**
	equipmentStatusCode	String
	itemTestId	String
Address		
	reference1	String
	reference2	String
	mailingName	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	zipPostalCode	String
	countryCode	String
PhoneNumber		
	areaCode	String
	phoneNumber	String
UserReserved		
	userReservedAmount	Integer
	userReservedCode	String
	userReservedReference	String
UserReservedChar		
	userReservedChar1	String
	userReservedChar2	String
UserReservedDate		
	userReservedDate	Calendar
	userReservedDate1	Calendar
	userReservedDate2	Calendar
UserReservedNumeric		
	userReservedNumeric1	Integer
	userReservedNumeric2	Integer

getCapitalAssetConditionBasedAlert – Response Interface (2 of 3)

ShowCapitalAssetConditionBasedAlert - Response Interface		
Class	**Field**	**Data Type**
UserReservedNumber		
	userReservedNumber	Integer
	userReservedNumber1	Integer
	userReservedNumber2	Integer
UserReservedString		
	userReservedString1	String
	userReservedString2	String
	userReservedString3	String
	userReservedString4	String
	userReservedString5	String
	userReservedString6	String
	userReservedString7	String
	userReservedString8	String
	userReservedString9	String
	userReservedString10	String
messages[]		
	messages	String

getCapitalAssetConditionBasedAlert – Response Interface (3 of 3)

Customer

This section lists these interface tables:

- getCustomer: Input Interface
- getCustomer: Response Interface
- getCustomerCreditInformation: Input Interface
- getCustomerCreditInformation: Response Interface
- processCustomer: Input Interface
- processCustomer: Response Interface

getCustomer: Input Interface

These tables list the input interface information for the getCustomer web service operation:

GetCustomer - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	entityName	String					
	entityTypeCode	String					
	businessUnit	String					
	company	String					
	industryClassificationCode	String					
	languageCode	String					
	customerBillingTypeCode	String					
Entity							
	entityId	Integer					Y
	entityLongId	String					
	entityTaxId	String					
AddressCustomer							
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
Financial							
	currencyCode	String					
	creditMessageCode	String					
Distribution							
GLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					
GLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					

getCustomer – Input Interface (1 of 3)

GetCustomer - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
ModelJournalEntry							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
CategoryCodesCustomer							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					
	categoryCode024	String					

getCustomer – Input Interface (2 of 3)

GetCustomer - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode025	String					
	categoryCode026	String					
	categoryCode027	String					
	categoryCode028	String					
	categoryCode029	String					
	categoryCode030	String					

getCustomer – Input Interface (3 of 3)

getCustomer: Response Interface

The getCustomer web service operation uses the ShowCustomer message as the response. These tables list the response interface information for the getCustomer web service operation:

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
CustomerResults[]		
	entityIdParent	Integer
	entityIdPayor	Integer
	entityIdDeductionManager	Integer
	entityTypeCode	String
	entityName	String
	languageCode	String
	company	String
	industryClassificationCode	String
	openedBy	String
	remark	String
Entity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
ShowAddress		
	mailingName	String
	mailingNameSecondary	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
	dateEffective	Calendar
CategoryCodesCustomer		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String

getCustomer – Response Interface (1 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
	categoryCode021	String
	categoryCode022	String
	categoryCode023	String
	categoryCode024	String
	categoryCode025	String
	categoryCode026	String
	categoryCode027	String
	categoryCode028	String
	categoryCode029	String
	categoryCode030	String
ShowDistribution		
	glOffSet	String
GLAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
ModelJournalEntry		
	documentNumber	Integer
	documentTypeCode	String

getCustomer – Response Interface (2 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	documentCompany	String
ShowFinancial		
	commissionCode1	Integer
	commissionCode2	BigDecimal
	rateCommissionCode1	Integer
	rateCommissionCode2	BigDecimal
Tax		
	taxRateAreaCode	String
	taxExplanationCode	String
	personCorporationCode	String
	entityTaxIdAdditional	String
ShowInvoice		
	paymentTermsCode	String
	paymentInstrumentCode	String
	currencyCode	String
	currencyCodeTransaction	String
	isInvoiceHeld	Boolean
	isAutoReceipt	Boolean
	autoReceiptExecutionList	String
	printAdjustments	String
	sendInvoiceToCPCode	String
Credit		
	amountCreditLimit	Integer
	administrationCreditLimit	Integer
	creditMessageCode	String
	creditManagerCode	String
	creditMessageCodeTemporary	String
	dunBradStreetRatingCode	String
	experianRatingCode	String
	ABCCodeSales	String
	ABCCodeMargin	String
	ABCCodeAverageDays	String
	dateExperianRating	Calendar
	dateDunBradStreetRating	Calendar

getCustomer – Response Interface (3 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	dateStatementReceived	Calendar
ShowCollection		
	collectionManagerCode	String
	policyNumber	Integer
	sendStatementToCode	String
	statementCycle	String
	isPrintStatementAllowed	Boolean
	printDeliquencyNotice	Boolean
	collectionReportCode	String
CustomerShipNotice		
	shippingLabelProgram	String
	processingVersion	String
	configurationDefault	String
	configurationPickPack	String
	configurationStandard	String
	isPackagingCodeRequired	Boolean
	isWeightCodeRequired	Boolean
	isTransportationMethodRequired	Boolean
	isRoutingDescriptionRequired	Boolean
	isEquipmentCodeRquired	Boolean
	isIdentificationCode1Required	Boolean
	isIdentificationCode2Required	Boolean
	isReference1Required	Boolean
	isReference2Required	Boolean
	defaultIdentificationCode1	String
	defaultIdentificationCode2	String
	defaultIdentificationCode3	String
	defaultReferenceCode	String
ShowBillingInstructions		
	entityIdCarrier	Integer
	entityIdRelated1	Integer
	customerBillingTypeCode	String
	adjustmentScheduleCode	String
	customerPriceGroupCode	String

getCustomer – Response Interface (4 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	orderValueMinimum	Integer
	orderValueMaximum	Integer
	creditCheckLevelCode	String
	discountTrade	BigDecimal
	orderTemplateCode	String
	itemRestrictionsCode	String
	deliveryInstruction1	String
	deliveryInstruction2	String
	numberOfInvoices	Integer
	holdCode	String
	priorityProcessingCode	String
	unitOfMeasureVolumeCode	String
	unitOfMeasureWeightCode	String
	isCustomerPORequired	Boolean
	zoneNumber	String
	segmentCodeBuying	String
	routeCode	String
	stopCode	String
	businessObjectiveCode	String
	isCreditHoldExempt	String
	freightHandlingCode	String
	isInvoiceAConsolidation	Boolean
	isPartialOrderShipmentAllowed	Boolean
	isPartialLineShipmentAllowed	Boolean
	isBackOrderAllowed	Boolean
	isSubstitutionAllowed	Boolean
	isFreightAllowed	Boolean
	printDeliveryNote	String
	isCustomerActive	Boolean
	bypassAuditLogging	String
	isPrintCOAAllowed	Boolean
UserReservedDataCustomer		
	userReservedCode	String
	userReservedDate	Calendar

getCustomer – Response Interface (5 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
ShowDates		
	dateCreditReviewLast	Calendar
	dateCreditReview	Calendar
	dateDunningLetterLast	Calendar
	dateStatementLast	Calendar
	dateInvoiceFirst	Calendar
	dateInvoiceLast	Calendar
	dateInvoiceNext	Calendar
	datePaidLast	Calendar
	dateOpen	Calendar
	dateLicenseExpire	Calendar
	datePricing	String
Amounts		
	amountDue	BigDecimal
	amountFinanceChargesPrior	BigDecimal
	amountFinanceChargesYTD	BigDecimal
	amountCreditLimit	Integer
	amountInvoicePrior	BigDecimal
	amountInvoiceCurrent	BigDecimal
	amountHighBalance	BigDecimal
	amountAppliedLast	BigDecimal
	amountAddressBook1	BigDecimal
	amountAddressBook2	BigDecimal
	amountOpen	BigDecimal
Miscellaneous		
	sequenceLedgerCode	String
	autoReceiptAlgorithmCode	String
	balanceForwardOpenCode	String
	creditCheckHandle	String
	creditReviewedLastBy	String
	daysSalesOutstanding	Integer

getCustomer – Response Interface (6 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	numberOfDunningLetters	Integer
	numberOfReminders	String
	numberOfSentReminders1	Integer
	numberOfSentReminders2	Integer
	numberOfSentReminders3	Integer
	numberOfInvoicesToCollectionReport	Integer
	financeChargeCode	String
	daysFinanceCharge	Integer
	percentageFactor	BigDecimal
	isFinanceChargeAllowed	Boolean
	cashReceiptsPending	String
	daysToPay	Integer
	customerIndustryGroup	Integer
	overrideTerritoryId	String
	specialInstruction4	String
	specialInstruction5	String
	preferenceSchedule	String
	percentPaidThreshold	Integer
	territoryId	Integer
	tierCode	String
	businessUnitAPS	String
	isShippingOfHeldLotsAllowed	String
	sendMethodCode	String
	palletControl	String
	quantityDecimals	Integer
	amountDecimals	Integer
	batchProcessingModeCode	String
	isPickPriceListAllowed	Boolean
	isMergeOrderAllowed	Boolean
	itemTypeIdentifierCode	String
	customerTypeIdentifierCode	String
	ordersYearToDate	Integer
	ordersPriorYear	Integer
	billingFrequencyCode	String

getCustomer – Response Interface (7 of 8)

ShowCustomer - Response Interface		
Class	**Field**	**Data Type**
	daysTransit	Integer
	overrideSalesTeamAssignment	String
	printMessage	String
PhoneNumber[]		
	contactId	Integer
	relatedPersonId	String
	phoneLineNumber	String
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddress[]		
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	Integer
Messages[]		

getCustomer – Response Interface (8 of 8)

getCustomerCreditInformation: Input Interface

This table lists the input interface information for the getCustomerCreditInformation web service operation:

getCustomerCreditInformation - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	currencyCode	String				Y	
	company	String				Y	
	businessUnit	String				Y	
entity							
	entityId	Integer				Y	Y
	entityLongId	String				Y	
	entityTaxId	String				Y	

getCustomerCreditInformation – Input Interface

getCustomerCreditInformation: Response Interface

The getCustomerCreditInformation web service operation uses the ShowCustomerCreditInformation message as the response. This table lists the response interface information for the getCustomerCreditInformation web service operation:

ShowCustomerCreditInformation - Response Interface		
Class	**Field**	**Data Type**
	amountTotalExposure	BigDecimal
	amountCreditLimit	BigDecimal
	isCreditHoldExempt	Boolean
entity		
	EntityId	Integer
	EntityLongId	String
	EntityTaxId	String
messages[]		
	messages	E1MessageList

getCustomerCreditInformation – Response Interface

processCustomer: Input Interface

These tables list the input interface information for the processCustomer web service operation:

processCustomer - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
Customer							
	company	String					
	openedBy	String					
	tierCode	String					
Processing							
	actionType	String	Y	Y	Y	Y	
	processingVersion	String					
AlternateAddressNumber - Entity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Parent - Entity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
AlternatePayor - Entity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
DeductionManager - Entity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Distribution							
	glOffSet	String					
GLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					

processCustomer – Input Interface (1 of 6)

processCustomer - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
GLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					
ModelJournalEntry							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
Invoice							
	paymentTermsCode	String					
	paymentInstrumentCode	String					
	currencyCode	String					
	currencyCodeTransaction	String					
	isInvoiceHeld	Boolean					
	isAutoReceipt	Boolean					
	autoReceiptExecutionList	String					
Financial							
	commissionCode1	Integer					
	commissionCode2	Integer					
Tax							
	taxRateAreaCode	String					
	taxExplanationCode	String					
	personCorporationCode	String					
	entityTaxIdAdditional	String					
Credit							
	amountCreditLimit	Integer					
	administrationCreditLimit	Integer					
	creditMessageCode	String					
	creditManagerCode	String					
	creditMessageCodeTemporary	String					

processCustomer – Input Interface (2 of 6)

processCustomer - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	dunBradStreetRatingCode	String					
	experianRatingCode	String					
	ABCCodeSales	String					
	ABCCodeMargin	String					
	ABCCodeAverageDays	String					
	dateExperianRating	Calendar					
	dateDunBradStreetRating	Calendar					
	dateStatementReceived	Calendar					
CategoryCodesCustomer							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					

processCustomer – Input Interface (3 of 6)

processCustomer - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode022	String					
	categoryCode023	String					
	categoryCode024	String					
	categoryCode025	String					
	categoryCode026	String					
	categoryCode027	String					
	categoryCode028	String					
	categoryCode029	String					
	categoryCode030	String					
CustomerShipNotice							
	shippingLabelProgram	String					
	processingVersion	String					
	configurationDefault	String					
	configurationPickPack	String					
	configurationStandard	String					
	isPackagingCodeRequired	Boolean					
	isWeightCodeRequired	Boolean					
	isTransportationMethodRequired	Boolean					
	isRoutingDescriptionRequired	Boolean					
	isEquipmentCodeRquired	Boolean					
	isIdentificationCode1Required	Boolean					
	isIdentificationCode2Required	Boolean					
	isReference1Required	Boolean					
	isReference2Required	Boolean					
	defaultIdentificationCode1	String					
	defaultIdentificationCode2	String					
	defaultIdentificationCode3	String					
	defaultReferenceCode	String					

processCustomer – Input Interface (4 of 6)

processCustomer - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
Collection							
	collectionManagerCode	String					
	policyNumber	Integer					
	policyCompany	String					
	sendStatementToCode	String					
	statementCycle	String					
	isPrintStatementAllowed	Boolean					
	printDeliquencyNotice	String					
	processDeliquencyFees	String					
	collectReport	String					
BillingInstructions							
	customerBillingTypeCode	String					
	adjustmentScheduleCode	String					
	customerPriceGroupCode	String					
	orderValueMinimum	Integer					
	orderValueMaximum	Integer					
	creditCheckLevelCode	String					
	discountTrade	BigDecimal					
	orderTemplateCode	String					
	itemRestrictionsCode	String					
	deliveryInstruction1	String					
	deliveryInstruction2	String					
	numberOfInvoices	Integer					
	holdCode	String					
	priorityProcessingCode	String					
	unitOfMeasureVolumeCode	String					
	unitOfMeasureWeightCode	String					
	isCustomerPORequired	Boolean					
	zoneNumber	String					
	segmentCodeBuying	String					

processCustomer – Input Interface (5 of 6)

processCustomer - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	routeCode	String					
	stopCode	String					
	businessObjectiveCode	String					
	isCreditHoldExempt	Boolean					
	freightHandlingCode	String					
	IsInvoiceAConsolidation	Boolean					
	isPartialOrderShipmentAllowed	Boolean					
	isPartialLineShipmentAllowed	Boolean					
	isBackOrderAllowed	Boolean					
	isSubstitutionAllowed	Boolean					
	isFreightAllowed	Boolean					
	printDeliveryNote	String					
	isCustomerActive	Boolean					
	bypassAuditLogging	String					
	isPrintCOAAllowed	Boolean					
	relatedAddress1	Integer					
Carrier							
	entityId	String					
	entityLongId	String					
	entityTaxId	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					
Dates							
	dateOpen	Calendar					
	dateCreditReview	Calendar					
	dateCreditReviewLast	Calendar					

processCustomer – Input Interface (6 of 6)

processCustomer: Response Interface

The processCustomer web service operation uses the ConfirmProcessCustomer message as the response. This table lists the response interface information for the processCustomer web service operation:

ConfirmProcessCustomer - Response Interface		
PSBF VO Field Name		Data Type
	company	String
messages[]		
	messages	String

processCustomer – Response Interface

Customer Service

This section lists these interface tables:

- getServiceOrder: Input Interface
- getServiceOrder: Response Interface
- processServiceOrder: Input Interface
- processServiceOrder: Response Interface
- getCommitmentDateTime: Input Interface
- getCommitmentDateTime: Response Interface

getServiceOrder: Input Interface

These tables list the input interface information for the getServiceOrder web service operation:

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
header							
	documentNumber	Integer					
	documentTypeCode	String					
	serviceTypeCode	String					
	description	String					
	description2	String					
	documentNumberParent	String					
	taxRateAreaCode	String					
	taxExplanationCode	String					
	paymentInstrumentCode	String					
customerInfo							
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
site							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
contact							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	reference2	String					
	entityNameContact	String					
	areaCode	String					
	phoneNumber	String					
equipment							
	assetId	Integer					

getServiceOrder – Input Interface (1 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	businessUnitAlternate	String					
	lotNumber	String					
	productModelCode	String					
	productFamilyCode	String					
	caseNumber	Integer					
	quantityOrdered	BigDecimal					
	unitOfMeasureCode	String					
planning							
	statusCodeOrder	String					
	hoursEstimatedDowntime	BigDecimal					
	hoursActualDowntime	BigDecimal					
	hoursEstimated	BigDecimal					
	hoursActual	BigDecimal					
	dateCommitment	Calendar					
	timeCommitment	Integer					
	responseTime	BigDecimal					
	dateEstimatedStart	Calendar					
	timeEstimatedStart	Integer					
	dateEstimatedEnd	Calendar					
	timeEstimatedEnd	Integer					
	dateActualStart	Calendar					
	timeActualStart	Integer					
	dateActualEnd	Calendar					
	timeActualEnd	Integer					

getServiceOrder – Input Interface (2 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	comment	String					
	percentComplete	BigDecimal					
	timeZoneCode	String					
	daylightSavingsRuleCode	String					
entitlement							
	checkEntitlement	String					
	coverageGroupCode	String					
	threshold	BigDecimal					
	coverageTypeCode	String					
	methodOfPricingCode	String					
assignment							
serviceProviderPrimary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceProviderSecondary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceTechnicianPrimary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceTechnicianSecondary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
originator							
	entityId	Integer					
	entityLongId	String					

getServiceOrder – Input Interface (3 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityTaxId	String					
	businessUnitLeadCraft	String					
classification							
categoryCodes							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	priorityCode	String					
	approvalTypeCode	String					
	workOrderStatusChange	String					
	messageNumber	String					
	reference1	String					
	languageCode	String					

getServiceOrder – Input Interface (4 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	geographicalRegionCode	String					
	countryCode	String					
	billTypeCode	String					
	routingTypeCode	String					
accounting							
estimatedCostOfWork							
	amountEstimatedLabor	BigDecimal					
	amountEstimatedMaterial	BigDecimal					
	amountEstimatedOther	BigDecimal					
	amountEstimated	BigDecimal					
actualCostOfWork							
	amountActualLabor	BigDecimal					
	amountaActualMaterial	BigDecimal					
	amountActualOther	BigDecimal					
	amountActual	BigDecimal					
gLAccount							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
gLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					
	subledgerInactiveCode	String					
	glCategoryCovered	String					
	glCategoryNonCovered	String					
	threshold						
	amountPrePayment	BigDecimal					
	methodOfPricingCode						
	currencyCode	String					

getServiceOrder – Input Interface (5 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	currencyCodeTo	String					
	currencyConversionRate	BigDecimal					
	currencyModeCode	String					
serviceContract							
	documentNumber	Integer					
	documentVersion	String					
	documentLineNumber	BigDecimal					
userReservedData							
	userReservedReference	String					
	userReservedDate	Calendar					
	userReservedNumber	Integer					
	userReservedCode	String					
	userReservedAmount	BigDecimal					
laborLineKey							
	documentNumber	Integer					
	laborLineNumber	BigDecimal					
	operationTypeCode	String					
	businessUnit	String					
	operationStatusCode	String					
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	description	String					

getServiceOrder – Input Interface (6 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	runMachine	BigDecimal					
	quantityEstimated	BigDecimal					
	quantityActual	BigDecimal					
	crewSize	BigDecimal					
	ratePiecework	BigDecimal					
	hoursLaborSetup	BigDecimal					
	hoursQueue	BigDecimal					
	hoursMove	BigDecimal					
	dateStart	Calendar					
	dateRequested	Calendar					
	dateCompletion	Calendar					
	dateBilled	Calendar					
	timeBasisCode	String					
	percentOverlap	BigDecimal					
	messageNumber	String					
relatedOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	operationNext	BigDecimal					
	resourceUnitsConsumed	BigDecimal					
	unitOfMeasureCapacity	String					
	jobCategory	String					
	competencyTypeCode	String					
	competencyCode	String					
	competencyLevelFrom	BigDecimal					
	competencyLevelTo	BigDecimal					
	percentCovered	BigDecimal					

getServiceOrder – Input Interface (7 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
serviceTechnician							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	dateActualStart	Calendar					
	adjustmentScheduleCodeService	String					
	methodOfPricingCode	String					
	costTypeCode	String					
	checkEntitlement	String					
	billableCode	String					
	unitBillableEstimated	BigDecimal					
	amountEstimated	BigDecimal					
	unitBillableActual	BigDecimal					
	amountBillableActual	BigDecimal					
	amountClaim	BigDecimal					
	adjustmentScheduleCodePrice	String					
	methodOfPricingCodePay	String					
	isPayable	Boolean					
	currencyCode	String					
	rateExchangePayable	BigDecimal					
	currencyModeCode	String					
	amountEstimatedPayable	BigDecimal					
	amountEstimatedPayment	BigDecimal					
	rateActualPayable	BigDecimal					
	amountPayment	BigDecimal					
	businessUnitCasualPart	String					
causalPart							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					

getServiceOrder – Input Interface (8 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	itemCustomer	String					
	itemFreeForm	String					
	lotNumberSupplier	String					
supplierRecoveryVendor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceOrderPartsLine							
	documentNumber	Integer					
	partsLineNumber	Integer					
	documentTypeCode	String					
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	description	String					
	description1	String					
	quantityEstimated	BigDecimal					
	quantityActual	BigDecimal					
	dateRequested	Calendar					
	lineTypeCode	String					
	unitOfMeasureCode	String					
	statusCodeMaterial	String					
	businessUnitComponent	String					
	location	String					
	lotNumber	String					
	costTypeCode	String					
	issueTypeCode	String					

getServiceOrder – Input Interface (9 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	amountEstimated	BigDecimal					
	amountExtendedCost	BigDecimal					
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceProvider							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	relatedOrderKey						
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	checkEntitlement	String					
	percentCovered	BigDecimal					
	billableCode	String					
	methodOfPricingCode	String					
	adjustmentScheduleCodeBillable	String					
	dateBilled	Calendar					
	unitBillableEstimated	BigDecimal					
	amountEstimatedBillable	BigDecimal					
	unitBillableActual	BigDecimal					
	amountBillableActual	BigDecimal					
	isPayable	Boolean					

getServiceOrder – Input Interface (10 of 11)

getServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	methodOfPricingCodePay	String					
	payForParts	Boolean					
	priceAdjustmentSchedule	String					
	datePaid	Calendar					
	currencyCode	String					
	rateExchangePayable	BigDecimal					
	currencyModeCode	String					
	unitPayableEstimated	BigDecimal					
	amountEstimatedPayable	BigDecimal					
	unitEstimatedPayableForeign	BigDecimal					
	amountEstimatedPayableForeign	BigDecimal					
	unitPayableActual	BigDecimal					
	amountActualPayable	BigDecimal					
	unitPayableActualForeign	BigDecimal					
	amountActualPayableForeign	BigDecimal					
	businessUnit	String					
	businessUnitCasualPart	String					
causalPart							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	lotNumberSupplier	String					
supplierRecoveryVendor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	returnPolicy	String					

getServiceOrder – Input Interface (11 of 11)

getServiceOrder: Response Interface

The getServiceOrder web service operation uses the ShowServiceOrder message as the response. These tables list the response interface information for the getServiceOrder web service operation:

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
showServiceOrder[]		
	documentNumber	Integer
	documentTypeCode	String
	serviceTypeCode	String
	description	String
	description2	String
	documentNumberParent	Integer
	taxRateAreaCode	String
	taxExplanationCode	String
	paymentInstrumentCode	String
customerInfo		
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
site		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
contact		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	reference2	String
	entityNameContact	String
	areaCode	String
	phoneNumber	String
equipment		
	assetId	Integer
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String

getServiceOrder – Response Interface (1 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	itemFreeForm	String
	businessUnitAlternate	String
	lotNumber	String
	productModelCode	String
	productFamilyCode	String
	caseNumber	Integer
	quantityOrdered	BigDecimal
	unitOfMeasureCode	String
planning		
	statusCodeOrder	String
	hoursEstimatedDowntime	BigDecimal
	hoursActualDowntime	BigDecimal
	hoursEstimated	BigDecimal
	hoursActual	BigDecimal
	dateCommitment	Calendar
	timeCommitment	Integer
	responseTime	BigDecimal
	dateEstimatedStart	Calendar
	timeEstimatedStart	Integer
	dateEstimatedEnd	Calendar
	timeEstimatedEnd	Integer
	dateActualStart	Calendar
	timeActualStart	Integer
	dateActualEnd	Calendar
	timeActualEnd	Integer
	comment	String
	percentComplete	BigDecimal
	timeZoneCode	String
	daylightSavingsRuleCode	String
entitlement		
	checkEntitlement	String
	coverageGroupCode	String
	threshold	BigDecimal
	coverageTypeCode	String

getServiceOrder – Response Interface (2 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	methodOfPricingCode	String
assignment		
serviceProviderPrimary		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
serviceProviderSecondary		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
serviceTechnicianPrimary		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
serviceTechnicianSecondary		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
originator		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	businessUnitLeadCraft	String
classification		
categoryCodes		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String

getServiceOrder – Response Interface (3 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	categoryCode010	String
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
	priorityCode	String
	approvalTypeCode	String
	workOrderStatusChange	String
	messageNumber	String
	reference1	String
	languageCode	String
	geographicalRegionCode	String
	countryCode	String
	billTypeCode	String
	routingTypeCode	String
accounting		
estimatedCostOfWork		
	amountEstimatedLabor	BigDecimal
	amountEstimatedMaterial	BigDecimal
	amountEstimatedOther	BigDecimal
	amountEstimated	BigDecimal
actualCostOfWork		
	amountActualLabor	BigDecimal
	amountaActualMaterial	BigDecimal
	amountActualOther	BigDecimal
	amountActual	BigDecimal
gLAccount		
	objectAccount	String

getServiceOrder – Response Interface (4 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	businessUnit	String
	subsidiary	String
	subledgerInactiveCode	String
	glCategoryCovered	String
	glCategoryNonCovered	String
	amountPrePayment	BigDecimal
	currencyCode	String
	currencyCodeTo	String
	currencyConversionRate	BigDecimal
	currencyModeCode	String
serviceContract		
	documentNumber	Integer
	documentVersion	String
	documentLineNumber	BigDecimal
userReservedData		
	userReservedReference	String
	userReservedDate	Calendar
	userReservedNumber	Integer
	userReservedCode	String
	userReservedAmount	BigDecimal
laborLineKey[]		
	documentNumber	Integer
	laborLineNumber	BigDecimal
	operationTypeCode	String
	businessUnit	String
	operationStatusCode	String
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String

getServiceOrder – Response Interface (5 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	itemCustomer	String
	itemFreeForm	String
	description	String
	runMachine	BigDecimal
	quantityEstimated	BigDecimal
	quantityActual	BigDecimal
	crewSize	BigDecimal
	ratePiecework	BigDecimal
	hoursLaborSetup	BigDecimal
	hoursQueue	BigDecimal
	hoursMove	BigDecimal
	dateStart	Calendar
	dateRequested	Calendar
	dateCompletion	Calendar
	dateBilled	Calendar
	timeBasisCode	String
	percentOverlap	BigDecimal
	messageNumber	String
relatedOrderKey		
	documentNumber	String
	documentTypeCode	String
	documentCompany	String
	documentLineNumber	BigDecimal
	operationNext	BigDecimal
	resourceUnitsConsumed	BigDecimal
	unitOfMeasureCapacity	String
	jobCategory	String
	competencyTypeCode	String
	competencyCode	String
	competencyLevelFrom	BigDecimal
	competencyLevelTo	BigDecimal
	percentCovered	BigDecimal
serviceTechnician		
	entityId	Integer

getServiceOrder – Response Interface (6 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	entityLongId	String
	entityTaxId	String
	dateActualStart	Calendar
	adjustmentScheduleCodeService	String
	methodOfPricingCode	String
	costTypeCode	String
	checkEntitlement	String
	billableCode	String
	unitBillableEstimated	BigDecimal
	amountEstimated	BigDecimal
	unitBillableActual	BigDecimal
	amountBillableActual	BigDecimal
	amountClaim	BigDecimal
	adjustmentScheduleCodePrice	String
	methodOfPricingCodePay	String
	isPayable	Boolean
	currencyCode	String
	rateExchangePayable	BigDecimal
	currencyModeCode	String
	amountEstimatedPayable	BigDecimal
	amountEstimatedPayment	BigDecimal
	rateActualPayable	BigDecimal
	amountPayment	BigDecimal
	businessUnitCausalPart	String
causalPart		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
	lotNumberSupplier	String
supplierRecoveryVendor		
	entityId	Integer
	entityLongId	String

getServiceOrder – Response Interface (7 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	entityTaxId	String
serviceOrderPartsLine[]		
	documentNumber - removed	Integer
	partsLineNumber	Integer
	documentTypeCode	String
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
	description	String
	description1	String
	quantityEstimated	BigDecimal
	quantityActual	BigDecimal
	dateRequested	Calendar
	lineTypeCode	String
	unitOfMeasureCode	String
	statusCodeMaterial	String
	businessUnitComponent	String
	location	String
	lotNumber	String
	costTypeCode	String
	issueTypeCode	String
	amountEstimated	BigDecimal
	amountExtendedCost	BigDecimal
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
supplier		
	entityId	Integer
	entityLongId	String
	entityTaxId	String

getServiceOrder – Response Interface (8 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
serviceProvider		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
relatedOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	documentLineNumber	BigDecimal
	checkEntitlement	String
	percentCovered	BigDecimal
	billableCode	String
	methodOfPricingCode	String
	adjustmentScheduleCodeBillable	String
	dateBilled	Calendar
	unitBillableEstimated	BigDecimal
	amountEstimatedBillable	BigDecimal
	unitBillableActual	BigDecimal
	amountBillableActual	BigDecimal
	isPayable	Boolean
	methodOfPricingCodePay	String
	payForParts	String
	priceAdjustmentSchedule	String
	datePaid	Calendar
	currencyCode	String
	rateExchangePayable	BigDecimal
	currencyModeCode	String
	unitPayableEstimated	BigDecimal
	amountEstimatedPayable	BigDecimal
	unitEstimatedPayableForeign	BigDecimal
	amountEstimatedPayableForeign	BigDecimal
	unitPayableActual	BigDecimal
	amountActualPayable	BigDecimal
	unitPayableActualForeign	BigDecimal

getServiceOrder – Response Interface (9 of 10)

ShowServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	amountActualPayableForeign	BigDecimal
	businessUnit	String
	businessUnitCausalPart	String
causalPart		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
	lotNumberSupplier	String
supplierRecoveryVendor		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	returnPolicy	String

getServiceOrder – Response Interface (10 of 10)

processServiceOrder: Input Interface

These tables list the input interface information for the processServiceOrder web service operation:

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
processHeader							
header							
	documentNumber	Integer		Y	Y		Y
	documentTypeCode	String					
	serviceTypeCode	String					
	description	String					
	description2	String	Y				
	documentNumberParent	String					
	taxRateAreaCode	String					
	taxExplanationCode	String					
	paymentInstrumentCode	String					
customerInfo							
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
site							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
contact							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	reference2	String					
	entityNameContact	String					
	areaCode	String					
	phoneNumber	String					
equipment							
	assetId	Integer					
item							
	itemId	Integer					

processServiceOrder – Input Interface (1 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	businessUnitAlternate	String					
	lotNumber	String					
	productModelCode	String					
	productFamilyCode	String					
	caseNumber	Integer					
	quantityOrdered	BigDecimal					
	unitOfMeasureCode	String					
planning							
	statusCodeOrder	String					
	hoursEstimatedDowntime	BigDecimal					
	hoursActualDowntime	BigDecimal					
	hoursEstimated	BigDecimal					
	hoursActual	BigDecimal					
	dateCommitment	Calendar					
	timeCommitment	Integer					
	responseTime	BigDecimal					
	dateEstimatedStart	Calendar					
	timeEstimatedStart	Integer					
	dateEstimatedEnd	Calendar					
	timeEstimatedEnd	Integer					
	dateActualStart	Calendar					
	timeActualStart	Integer					
	dateActualEnd	Calendar					
	timeActualEnd	Integer					
	comment	String					
	percentComplete	BigDecimal					
	timeZoneCode	String	Y				
	daylightSavingsRuleCode	String					

processServiceOrder – Input Interface (2 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
entitlement							
	checkEntitlement	String					
	coverageGroupCode	String					
	threshold	BigDecimal					
	coverageTypeCode	String					
	methodOfPricingCode	String					
assignment							
serviceProviderPrimary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceProviderSecondary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceTechnicianPrimary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceTechnicianSecondary							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
originator							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	businessUnitLeadCraft	String					
classification							
categoryCodes							
	categoryCode001	String					
	categoryCode002	String					

processServiceOrder – Input Interface (3 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	priorityCode	String					
	approvalTypeCode	String					
	workOrderStatusChange	String					
	messageNumber	String					
	reference1	String					
	languageCode	String					
	geographicalRegionCode	String					
	countryCode	String					
	billTypeCode	String					
	routingTypeCode	String					
accounting							
estimatedCostOfWork							
	amountEstimatedLabor	BigDecimal					
	amountEstimatedMaterial	BigDecimal					

processServiceOrder – Input Interface (4 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	amountEstimatedOther	BigDecimal					
	amountEstimated	BigDecimal					
actualCostOfWork							
	amountActualLabor	BigDecimal					
	amountaActualMaterial	BigDecimal					
	amountActualOther	BigDecimal					
	amountActual	BigDecimal					
gLAccount							
	objectAccount	String					
	businessUnit	String	Y				
	subsidiary	String					
gLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					
	subledgerInactiveCode	String					
	glCategoryCovered	String					
	glCategoryNonCovered	String					
	amountPrePayment	BigDecimal					
	currencyCode	String					
	currencyCodeTo	String					
	currencyConversionRate	BigDecimal					
	currencyModeCode	String					
serviceContract							
	documentNumber	Integer					
	documentVersion	String					
	documentLineNumber	BigDecimal					
userReservedData							
	userReservedReference	String					
	userReservedDate	Calendar					
	userReservedNumber	Integer					
	userReservedCode	String					

processServiceOrder – Input Interface (5 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	userReservedAmount	BigDecimal					
serviceOrderProcessing							
	actionType	String					
	processingVersion	String					
processLaborLineKey[]							
	actionType	String	Y	Y	Y		
laborLineKey							
	documentNumber	Integer	Y	Y	Y		Y
	laborLineNumber	BigDecimal		Y	Y		Y
	operationTypeCode	String	Y	Y	Y		Y
	businessUnit	String	Y	Y	Y		Y
	operationStatusCode	String					
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	description	String					
	runMachine	BigDecimal					
	quantityEstimated	BigDecimal					
	quantityActual	BigDecimal					
	crewSize	BigDecimal					
	ratePiecework	BigDecimal					
	hoursLaborSetup	BigDecimal					
	hoursQueue	BigDecimal					
	hoursMove	BigDecimal					
	dateStart	Calendar					

processServiceOrder – Input Interface (6 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	dateRequested	Calendar					
	dateCompletion	Calendar					
	dateBilled	Calendar					
	timeBasisCode	String					
	percentOverlap	BigDecimal					
	messageNumber	String					
relatedOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	operationNext	BigDecimal					
	resourceUnitsConsumed	BigDecimal					
	unitOfMeasureCapacity	String					
	jobCategory	String					
	competencyTypeCode	String					
	competencyCode	String					
	competencyLevelFrom	BigDecimal					
	competencyLevelTo	BigDecimal					
	percentCovered	BigDecimal					
serviceTechnician							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	dateActualStart	Calendar					
	adjustmentScheduleCodeService	String					
	methodOfPricingCode	String					
	costTypeCode	String					
	checkEntitlement	String					
	billableCode	String					
	unitBillableEstimated	BigDecimal					
	amountEstimated	BigDecimal					

processServiceOrder – Input Interface (7 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	unitBillableActual	BigDecimal					
	amountBillableActual	BigDecimal					
	amountClaim	BigDecimal					
	adjustmentScheduleCodePrice	String					
	methodOfPricingCodePay	String					
	isPayable	Boolean					
	currencyCode	String					
	rateExchangePayable	BigDecimal					
	currencyModeCode	String					
	amountEstimatedPayable	BigDecimal					
	amountEstimatedPayment	BigDecimal					
	rateActualPayable	BigDecimal					
	amountPayment	BigDecimal					
	businessUnitCasualPart	String					
causalPart							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	lotNumberSupplier	String					
supplierRecoveryVendor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
processServiceOrderPartsLine							
	actionType	String	Y	Y	Y	Y	
serviceOrderPartsLine							
	documentNumber	Integer					
	partsLineNumber	Integer		Y	Y		Y
	documentTypeCode	String					

processServiceOrder – Input Interface (8 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	description	String					
	description1	String					
	quantityEstimated	BigDecimal					
	quantityActual	BigDecimal					
	dateRequested	Calendar					
	lineTypeCode	String					
	unitOfMeasureCode	String					
	statusCodeMaterial	String					
	businessUnitComponent	String					
	location	String					
	lotNumber	String					
	costTypeCode	String					
	issueTypeCode	String					
	amountEstimated	BigDecimal					
	amountExtendedCost	BigDecimal					
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceProvider							
	entityId	Integer					
	entityLongId	String					

processServiceOrder – Input Interface (9 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityTaxId	String					
relatedOrderKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	checkEntitlement	String					
	percentCovered	BigDecimal					
	billableCode	String					
	methodOfPricingCode	String					
	adjustmentScheduleCodeBillable	String					
	dateBilled	Calendar					
	unitBillableEstimated	BigDecimal					
	amountEstimatedBillable	BigDecimal					
	unitBillableActual	BigDecimal					
	amountBillableActual	BigDecimal					
	isPayable	Boolean					
	methodOfPricingCodePay	String					
	payForParts	Boolean					
	priceAdjustmentSchedule	String					
	datePaid	Calendar					
	currencyCode	String					
	rateExchangePayable	BigDecimal					
	currencyModeCode	String					
	unitPayableEstimated	BigDecimal					
	amountEstimatedPayable	BigDecimal					
	unitEstimatedPayableForeign	BigDecimal					
	amountEstimatedPayableForeign	BigDecimal					
	unitPayableActual	BigDecimal					
	amountActualPayable	BigDecimal					
	unitPayableActualForeign	BigDecimal					
	amountActualPayableForeign	BigDecimal					

processServiceOrder – Input Interface (10 of 11)

processServiceOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	businessUnit	String					
	businessUnitCasualPart	String					
causalPart							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	lotNumberSupplier	String					
supplierRecoveryVendor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	returnPolicy	String					

processServiceOrder – Input Interface (11 of 11)

processServiceOrder: Response Interface

The processServiceOrder web service operation uses the ConfirmServiceOrder message as the response. These tables list the response interface information for the processServiceOrder web service operation:

ConfirmServiceOrder - Response Interface		
Class	**Field**	**Data Type**
confirmHeader		**ConfirmHeader**
	actionType	String
	documentNumber	Integer
	serviceTypeCode	String
	documentTypeCode	String
	priorityCode	String
	description	String
	description2	String
	documentNumberParent	String
	taxRateAreaCode	String
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
site		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
confirmLaborLineKey[]		
	documentNumber	Integer
	laborLineNumber	BigDecimal
	operationTypeCode	String
	businessUnit	String
	actionType	String
	description	String
	serviceTechnician	Integer
	quantityEstimated	BigDecimal
	quantityActual	BigDecimal
	unitOfMeaasureCapacity	String
	unitEstimatedBillable	BigDecimal
	amountEstimated	BigDecimal
confirmServiceOrderPartsLine[]		
	partsLineNumber	Integer
	item	Integer

processServiceOrder – Response Interface (1 of 2)

ConfirmServiceOrder - Response Interface		
Class	**Field**	**Data Type**
	quantityEstimated	BigDecimal
	quanittyActual	BigDecimal
	unitOfMeasureCode	String
	businessUnitComponent	String
messages[]		

processServiceOrder – Response Interface (2 of 2)

getCommitmentDateTime: Input Interface

This table lists the input interface information for the getCommitmentDateTime web service operation:

getCommitmentDateTime - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
customer							
	entityId	Integer				Y	Y
	entityLongId	String				Y	
	entityTaxId	String				Y	
site							
	entityId	Integer				Y	
	entityLongId	String				Y	
	entityTaxId	String				Y	
	coverageType	String				Y	
	lineTypeCode	String				Y	
	dateActualStart	Calendar				Y	
	timeActualSart	Integer				Y	
	responseTime	Integer				Y	
	threshold	Integer				Y	

getCommitmentDateTime – Input Interface

getCommitmentDateTime: Response Interface

The getCommitmentDateTime web service operation uses the ShowCommitmentDateTime message as the response. This table lists the response interface information for the getCommitmentDateTime web service operation:

ShowCommitmentDateTime - Response Interface		
Class	Field	Data Type
	dateActualStart	Calendar
	timeActualStart	Integer
	dateCommitment	Calendar
	timeCommitment	Integer
messages[]		
	messages	String

getCommitmentDateTime – Response Interface

Equipment

This section lists these interface tables:

- equipmentProcessor: Input Interface
- equipmentProcessor: Response Interface
- equipmenetQueryProcessor: Input Interface
- equipmentQueryProcessor: Response Interface

equipmentProcessor: Input Interface

These tables list the input interface information for the equipmentProcessor web service operation:

EquipmentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Processing							
	actionType	String	Y	Y			
	processingVersion	String					
EquipmentRecord							
	businessUnitLocation	String					
	sequenceNumber	Integer					
	descriptionCompressed	String					
	quantityAssetItemCurrent	BigDecimal					
	quantityAssetItemOriginal	BigDecimal					
	isInvestmentTaxCredit	Boolean					
	standardFuelConsumption	String					
	jobCategory	String					
	jobStep	String					
	unionCode	String					
	criticalityCode	String					
	updateChildBusinessUnitOnChange	String					
	reference	String					
	vehicleIdentificationNumber	String					
	openedBy	String					
CustomerInformation							
Customer							
	entityId	Integer	Y	Y			
	entityLongId	String					
	entityTaxId	String					
Site							
	entityId	Integer	Y	Y			
	entityLongId	String					
	entityTaxId	String					
	countryCode	String					
	isTermsAccepted	Boolean					
	registrationStatusCode	String					

equipmentProcessor – Input Interface (1 of 8)

EquimentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Equipment Information							
	assetId	Integer	Y	Y			
	description	String	Y	Y			
	unitNumber	String	Y	Y			
	serialNumber	String	Y	Y			
	assetIdParent	Integer					
	businessUnit	Integer					
	statusEquipmentCode	String					
	isAssetOwned	Boolean					
	productModelCode	String					
	productFamilyCode	Integer					
	description1	String					
	description2	String					
	salesTypeCode	String					
	partsNumber	Integer					
	isMeterReadingRequired	Boolean					
	isWorkOrderAllowed	Boolean					
	proofOfPurchaseCode	String					
Item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
Equipmentdates							
	dateAcquired	Calendar					
	dateDisposition	Calendar					
	dateInstallation	Calendar					
	dateInService	Calendar					
Accounting							
GLAccount							
	objectAccount	String		Y			

equipmentProcessor – Input Interface (2 of 8)

EquipmentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	businessUnit	String					
	subsidiary	String					
	company	String	Y	Y			Y
	subledgerInactiveCode	String					
	afeNumber	String					
StatusHistory							
	dateChange	Calendar					
	timeChange	String					
	dateLastChanged	Calendar					
	changeChildrenStatus	String					
	changeChildrenCustomerNumber	String					
Financing							
	purchaseOptionCode	String					
	amountPurchaseOption	BigDecimal					
	amountCreditLimit	BigDecimal					
	percentPurchaseOptionCredit	BigDecimal					
	dateExpiration	Calendar					
	amountPayment	BigDecimal					
	description1	String					
	description2	String					
ShipTo							
	entityId	Integer	Y	Y			Y
	entityLongId	String					
	entityTaxId	String					
Insurance							
	valueIndexInsurance	Integer					
	amountInsurance	BigDecimal					
	amountMonthlyPayment	BigDecimal					
	amountInsurancePremium	BigDecimal					
	company	String					
	policyNumber	String					
	monthPolicyRenewal	String					

equipmentProcessor – Input Interface (3 of 8)

EquimentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	costReplacement	BigDecimal					
	costReplacementLastYear	BigDecimal					
Depreciation							
GLAccountAccumulated							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
GLAccountDepreciationExpense							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
GLAccountRevenue							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
GLAccountDisposal							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
	amountEstimatedSalvageValue	BigDecimal					
Service							
ServiceProvider							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
ServiceTechnician							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
ServiceDates							
	dateExpectedReturn	Calendar					
	DateEffective	Calendar					

equipmentProcessor – Input Interface (4 of 8)

EquipmentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
warrantyInformation							
	dateExpiration	Calendar					
SalesPerson							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Distributor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
DefaultDealer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Assessor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
CategoryCodesAsset							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					

equipmentProcessor – Input Interface (5 of 8)

EquimentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					
CategoryCodesEquipment							
	productComponent	String					
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
AdditionalInformation							
Remarks							
	description1	String					
	description2	String					
	description3	String					
TaxInformation							
TaxEntity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

equipmentProcessor – Input Interface (6 of 8)

EquimentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	newOrUsedCode	String					
	financingMethodCode	String					
	stateCode	String					
	amountTax	BigDecimal					
RelatedOrder							
Sales							
SalesOrderLineKey							
	documentNumber	BigDecimal					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	Integer					
	dateOrdered	Calendar					
	dateShipment	Calendar					
	dateActualShip	Calendar					
	location	String					
	lotNumber	String					
Manufacturing							
RelatedOrderNumber							
	documentNumber	String					
	documentTypeCode	String					
	dateActualEnd	Calendar					
WorkOrder							
	documentNumber	Integer					
	billRevisionLevel	String					
	dateEffective	Calendar					
	businessUnit	String					
OriginalOrderNumber							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentSuffix	String					
	documentLineNumber	BigDecimal					

equipmentProcessor – Input Interface (7 of 8)

EquimentProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
PrimaryLastVendor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
UserReservedData							
	userReservedReference	String					
	userReservedDate	Calendar					
	userReservedNumber	Integer					
	userReservedCode	String					
	userReservedAmount	Integer					

equipmentProcessor – Input Interface (8 of 8)

equipmentProcessor: Response Interface

The equipmentProcessor web service operation uses the ConfirmEquipment message as the response. This table lists the response interface information for the equipmentProcessor web service operation:

ConfirmEquipment- Response Interface		
Class	**Field**	**Data Type**
Equipment		
	equipment	Integer
	SerialNumber	String
	unitNumber	String
	item	Integer
	description	String
	productModel	String
	productFamily	String
	businessUnit	Integer
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
site		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
messages[]		
	messages	String

equipmentProcessor – Response Interface

equipmentQueryProcessor: Input Interface

These tables list the input interface information for the equipmentQueryProcessor web service operation:

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
equipment							
customerInformation							
customer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
site							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	countryCode	String					
	isTermsAccepted	Boolean					
	registrationStatusCode	String					
equipmentInformation							
	assetId	Integer					
	description	String					
	unitNumber	String					
	serialNumber	String					
	assetIdParent	Integer					
	businessUnit	String					
	statusEquipmentCode	String					
	isAssetOwned	Boolean					
	productModelCode	String					
	productFamilyCode	String					
	description1	String					
	description2	String					
	salesTypeCode	String					
	partsNumber	Integer					
	isMeterReadingRequired	Boolean					
	isWorkOrderAllowed	Boolean					
	proofOfPurchaseCode	String					

EquipmentQueryProcessor – Input Interface (1 of 7)

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
equipmentDates							
	dateAcquired	Calendar					
	dateDisposition	Calendar					
	dateInstallation	Calendar					
	dateInService	Calendar					
accounting							
gLAccount							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
	company	String					
	subledgerInactiveCode	String					
	afeNumber	String					
statusHistory							
	dateChange	Calendar					
	dateLastChanged	Calendar					
financing							
	purchaseOptionCode	String					
	amountPurchaseOption	BigDecimal					
	amountCreditLimit	BigDecimal					
	percentPurchaseOptionCredit	BigDecimal					
	dateExpiration	Calendar					
	amountPayment	BigDecimal					
	description1	String					
	description2	String					

EquipmentQueryProcessor – Input Interface (2 of 7)

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
shipTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
insurance							
	valueIndexInsurance	BigDecimal*					
	amountInsurance	BigDecimal					
	amountInsurancePremium	BigDecimal					
	company	String					
	policyNumber	String					
	monthPolicyRenewal	Integer					
	costReplacement	BigDecimal					
	costReplacementLastYear	BigDecimal					
depreciation							
gLAccountAccumulated							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
gLAccountDepreciationExpense							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
gLAccountRevenue							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					
gLAccountDisposal							
	objectAccount	String					
	businessUnit	String					
	subsidiary	String					

EquipmentQueryProcessor – Input Interface (3 of 7)

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	amountEstimatedSalvageValue	BigDecimal					
service							
serviceProvider							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceTechnician							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
serviceDates							
	dateExpectedReturn	Calendar					
	DateEffective	Calendar					
warrantyInformation							
	dateExpiration	Calendar					
salesPerson							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
distributor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
defaultDealer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
assessor							
	entityId	Integer					
	entityLongId	String					

EquipmentQueryProcessor – Input Interface (4 of 7)

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityTaxId	String					
categoryCodesAsset							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					
categoryCodesEquipment							
	productComponent	String					
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					

EquipmentQueryProcessor – Input Interface (5 of 7)

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
additionalInformation							
remarks							
	description1	String					
	description2	String					
	description3	String					
taxInformation							
taxEntity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	newOrUsedCode	String					
	financingMethodCode	String					
	stateCode	String					
	amountTax	BigDecimal					
relatedOrder							
sales							
salesOrderLineKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	dateOrdered	Calendar					
	dateShipment	Calendar					

EquipmentQueryProcessor – Input Interface (6 of 7)

EquipmentQueryProcessor - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	dateActualShip	Calendar					
	location	String					
	lotNumber	String					
manufacturing							
relatedOrderNumber							
	documentNumber	String					
	documentTypeCode	String					
	dateEffective	Calendar					
	dateActualEnd	Calendar					
workOrder							
	documentNumber	Integer					
	billRevisionLevel	String					
	businessUnit	String					
originalOrderNumber							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentSuffix	String					
	documentLineNumber	BigDecimal					
primaryLastVendor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
userReservedData							
	userReservedReference	String					
	userReservedDate	Calendar					
	userReservedNumber	Integer					
	userReservedCode	String					
	userReservedAmount	BigDecimal					

EquipmentQueryProcessor – Input Interface (7 of 7)

equipmentQueryProcessor: Response Interface

The equipmentQueryProcessor web service operation uses the ShowEquipment message as the response. These tables list the response interface information for the equipmentQueryProcessor web service operation:

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
equipment		
customerInformation		
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
site		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	countryCode	String
	isTermsAccepted	Boolean
	registrationStatusCode	String
equipmentInformation		
	assetId	Integer
	description	String
	unitNumber	String
	serialNumber	String
	assetIdParent	Integer
	businessUnit	String
	statusEquipmentCode	String
	isAssetOwned	Boolean
	productModelCode	String
	productFamilyCode	String
	description1	String
	description2	String
	salesTypeCode	String
	partsNumber	Integer
	isMeterReadingRequired	Boolean
	isWorkOrderAllowed	Boolean
	proofOfPurchaseCode	String
item		

EquipmentQueryProcessor – Response Interface (1 of 7)

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
equipmentDates		
	dateAcquired	Calendar
	dateDisposition	Calendar
	dateInstallation	Calendar
	dateInService	Calendar
	businessUnitLocation	String
	sequenceNumber	BigDecimal*
	descriptionCompressed	String
	quantityAssetItemCurrent	BigDecimal
	quantityAssetItemOriginal	BigDecimal
	isInvestmentTaxCredit	Boolean
	standardFuelConsumption	BigDecimal
	jobCategory	String
	jobStep	String
	unionCode	String
	criticalityCode	Integer
	reference	String
	vehicleIdentificationNumber	String
	openedBy	String
accounting		
gLAccount		
	objectAccount	String
	businessUnit	String
	subsidiary	String
	company	String
	subledgerInactiveCode	String
	afeNumber	String

EquipmentQueryProcessor – Response Interface (2 of 7)

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
statusHistory		
	dateChange	Calendar
	dateLastChanged	Calendar
financing		
	purchaseOptionCode	String
	amountPurchaseOption	BigDecimal
	amountCreditLimit	BigDecimal
	percentPurchaseOptionCredit	BigDecimal
	dateExpiration	Calendar
	amountPayment	BigDecimal
	description1	String
	description2	String
	shipToEntityId	Integer
insurance		
	valueIndexInsurance	BigDecimal
	amountInsurance	BigDecimal
	amountInsurancePremium	BigDecimal
	company	String
	policyNumber	String
	monthPolicyRenewal	Integer
	costReplacement	BigDecimal
	costReplacementLastYear	BigDecimal
depreciation		
gLAccountAccumulated		
	objectAccount	String
	businessUnit	String
	subsidiary	String
gLAccountDepreciationExpense		
	objectAccount	String
	businessUnit	String
	subsidiary	String

EquipmentQueryProcessor – Response Interface (3 of 7)

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
gLAccountRevenue		
	objectAccount	String
	businessUnit	String
	subsidiary	String
gLAccountDisposal		
	objectAccount	String
	businessUnit	String
	subsidiary	String
	amountEstimatedSalvageValue	BigDecimal
service		
	serviceProviderEntityId	Integer
	serviceTechnicianEntityId	Integer
serviceDates		
	dateExpectedReturn	Calendar
	DateEffective	Calendar
warrantyInformation		
	dateExpiration	Calendar
	salesPersonEntityId	Integer
	distributorEntityId	Integer
	defaultDealerEntityId	Integer
	assessorEntityId	Integer
categoryCodesAsset		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String

EquipmentQueryProcessor – Response Interface (4 of 7)

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
	categoryCode021	String
	categoryCode022	String
	categoryCode023	String
categoryCodesEquipment		
	productComponent	String
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
additionalInformation		
remarks		
	description1	String
	description2	String
	description3	String

EquipmentQueryProcessor – Response Interface (5 of 7)

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
taxInformation		
taxEntity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	newOrUsedCode	String
	financingMethodCode	String
	stateCode	String
	amountTax	BigDecimal
relatedOrder		
sales		
salesOrderLineKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	documentLineNumber	BigDecimal
	dateOrdered	Calendar
	dateShipment	Calendar
	dateActualShip	Calendar
	location	String
	lotNumber	String
manufacturing		
relatedOrderNumber		
	documentNumber	String
	documentTypeCode	String
	dateEffective	Calendar
	dateActualEnd	Calendar
workOrder		
	documentNumber	Integer
	billRevisionLevel	String
	businessUnit	String

EquipmentQueryProcessor – Response Interface (6 of 7)

ShowEquipment - Response Interface		
Class	**Field**	**Data Type**
originalOrderNumber		
	documentNumber	String
	documentTypeCode	String
	documentCompany	String
	documentSuffix	String
	documentLineNumber	BigDecimal
primaryLastVendor		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
userReservedData		
	userReservedReference	String
	userReservedDate	Calendar
	userReservedNumber	Integer
	userReservedCode	String
	userReservedAmount	BigDecimal
messages[]		
	messages	String

EquipmentQueryProcessor – Response Interface (7 of 7)

Financial Compliance

This section lists these interface tables:

- getAPProcessingOptions: Input Interface
- getAPProcessingOptions: Response Interface
- getWriterOffProcessingOptions: Input Interface
- getWriterOffProcessingOptions: Response Interface
- getAgingCompanyConstants: Input Interface
- getAgingCompanyConstants: Response Interface
- getCustomerCreditLimits: Input Interface
- getCustomerCreditLimits: Response Interface
- getGeneralConstants: Input Interface
- getGeneralConstants: Response Interface
- getJournalGeneralConstants: Input Interface
- getJournalGeneralConstants: Response Interface
- getPolicyEditRules: Input Interface
- getPolicyEditRules: Response Interface
- getPurhcasingToleranceRules: Input Interface

- getPurhcasingToleranceRules: Response Interface
- getSecurityWorkbench: Input Interface
- getSecurityWorkbench: Response Interface

getAPProcessingOptions: Input Interface

This table lists the input interface information for the getAPProcessingOptions web service operation:

getAPProcessingOptions - Input Interface			Required				Key
Class	Fields	Data Type	A	C	D/C	I	(Y/N)
	processingVersionRecycleRecurringVouchers[]	String					
	processingVersionPOReceipts[]	String					

getAPProcessingOptions – Input Interface

getAPProcessingOptions: Response Interface

The getAPProcessingOptions web service operation uses the ShowAPProcessingOptions message as the response. This table lists the response interface information for the getAPProcessingOptions web service operation:

ShowAPProcessingOptions - Response Interface		
Class	Field	Data Type
RecycleRecurringVouchers[]		
	invoiceNumberFlag	String
	processingVersionRecycleRecurringVouchers	String
POReceipts[]		
	toleranceOnQuantityAmount	String
	toleranceOnDate	String
	processingVersionPOReceipts	String
Messages[]		

getAPProcessingOptions – Response Interface

getWriterOffProcessingOptions: Input Interface

This table lists the input interface information for the getWriterOffProcessingOptions web service operation:

getWriteOffProcessingOptions - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	processingVersionStandardReceiptEntry[]	String					
	processingVersionDraftEntry[]	String					
	processingVersionSpeedReceiptsEntry[]	String					
	processingVersionInvoiceSelectionMatch[]	String					
	processingVersionKnownInvoiceMatchWithAmount[]	String					
	processingVersionKnownInvoiceMatchWithoutAmount[]	String					

getWriterOffProcessingOptions – Input Interface

getWriterOffProcessingOptions: Response Interface

The getWriterOffProcessingOptions web service operation uses the ShowWriteOffProcessingOptions message as the response. This table lists the response interface information for the getWriterOffProcessingOptions web service operation:

ShowWriteOffProcessingOptions - Response Interface		
Class	**Field**	**Data Type**
standardReceiptEntry[]		
	autoWriteOffMaximumUnderpaymentAmount	String
	autoWriteOffMaximumOverpaymentAmount	String
	manualWriteOffmaximumUnderpaymentAmount	String
	manualWriteOffMaximumOverpaymentAmount	String
	versionStandardReceiptEntry	String
draftEntry[]		
	autoWriteOffMaximumUnderpaymentAmount	String
	autoWriteOffMaximumOverpaymentAmount	String
	manualWriteOffmaximumUnderpaymentAmount	String
	manualWriteOffMaximumOverpaymentAmount	String
	versionDraftEntry	String
speedReceiptsEntry[]		
	autoWriteOffMaximumUnderpaymentAmount	String
	autoWriteOffMaximumOverpaymentAmount	String
	versionSpeedReceiptsEntry	String
invoiceSelectionMatch[]		
	matchingUnderpaidToleranceAmount	String
	matchingOverpaidToleranceAmount	String
	versionInvoiceSelectionMatch	String
knownInvoiceMatchWithAmount[]		
	applyInvoicesUnderpaidToleranceAmount	String
	applyInvoicesOverpaidToleranceAmount	String
	applyReceiptsUnderpaidToleranceAmount	String
	applyReceiptsOverpaidToleranceAmount	String
	versionKnownInvoiceMatchWithAmount	String
knownInvoiceMatchWithoutAmount[]		
	applyReceiptsUnderpaidToleranceAmount	String
	applyReceiptsOverpaidToleranceAmount	String
	versionKnownInvoiceMatchWithoutAmount	String

getWriterOffProcessingOptions – Response Interface

getAgingCompanyConstants: Input Interface

This table lists the input interface information for the getAgingCompanyConstants web service operation:

getAgingCompanyConstants - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	fromCompany	String					Y
	toCompany	String					Y

getAgingCompanyConstants – Input Interface

getAgingCompanyConstants: Response Interface

The getAgingCompanyConstants web service operation uses the ShowAgingCompanyConstants message as the response. This table lists the response interface information for the getAgingCompanyConstants web service operation:

ShowAgingCompanyConstants - Response Interface		
Class	**Field**	**Data Type**
agingCompanyConstants[]		
	company	String
	dateAgeAsOf	Calendar
	agingMethod	String
	agingDateBasedOn	String
	agingDaysAR1	Integer
	agingDaysAR2	Integer
	agingDaysAR3	Integer
	agingDaysAR4	Integer
	agingDaysAR5	Integer
	agingDaysAR6	Integer
	agingDaysAR7	Integer
	currencyConversion	String
	agingDaysARCurrent	Intenger
Messages[]		

getAgingCompanyConstants – Response Interface

getCustomerCreditLimits: Input Interface

This table lists the input interface information for the getCustomerCreditLimits web service operation:

getCustomerCreditLimits - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	fromCustomer	Integer					
	toCustomer	Integer					

getCustomerCreditLimits – Input Interface

getCustomerCreditLimits: Response Interface

The getCustomerCreditLimits web service operation uses the ShowCustomerCreditLimits message as the response. This table lists the response interface information for the getCustomerCreditLimits web service operation:

ShowCustomerCreditLimits - Response Interface		
Class	**Field**	**Data Type**
CustomerCreditLimits []		
	entityId	Integer
	company	String
	amountCreditLimit	Integer
	dateCreditReview	Calendar
Messages[]		

getCustomerCreditLimits – Response Interface

getGeneralConstants: Input Interface

This table lists the input interface information for the getGeneralConstants web service operation:

getGeneralConstants - Input Interface			Required				Key
PSBF VO Field Name		Data Type	A	C	D/C	I	(Y/N)
	compliance	String					

getGeneralConstants – Input Interface

getGeneralConstants: Response Interface

The getGeneralConstants web service operation uses the ShowGeneralConstants message as the response. This table lists the response interface information for the getGeneralConstants web service operation:

ShowGeneralConstants - Response Interface		
Class	Field	Data Type
	duplicateInvoiceNumberEdit	String
Messages[]		

getGeneralConstants – Response Interface

getJournalGeneralConstants: Input Interface

This table lists the input interface information for the getJournalGeneralConstants web service operation:

getJournalGeneralConstants - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	compliance	String					

getJournalGeneralConstants – Input Interface

getJournalGeneralConstants: Response Interface

The getJournalGeneralConstants web service operation uses the ShowGeneralJournalConstants message as the response. This table lists the response interface information for the getJournalGeneralConstants web service operation:

ShowJournalGeneralConstants - Response Interface		
Class	Field	Data Type
	isBatchControlRequired	Boolean
	isBatchManagementApprovalRequired	Boolean
	arePostingsAllowed	Boolean
	areInvalidAccountsAllowed	Boolean
	intercompanySettlements	String
	isMultiCurrencyIntercompanyTransAllowed	Boolean
Messages[]		

getJournalGeneralConstants – Response Interface

getPolicyEditRules: Input Interface

This table lists the input interface information for the getPolicyEditRules web service operation:

getPolicyEditRules - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	compliance	String					

getPolicyEditRules – Input Interface

getPolicyEditRules: Response Interface

The getPolicyEditRules web service operation uses the ShowPolicyEditRules message as the response. This table lists the response interface information for the getPolicyEditRules web service operation:

ShowPolicyEditRules - Response Interface		
Class	Field	Data Type
PolicyEditRules []		
	dateEffective	Calendar
	policyName	String
	hardEdit	String
	preferredSupplier	String
	dailyAllowance	BigDecimal
	receiptRequired	String
	policyCurrencyCode	String
	expenseCategory	String
	expenseReportType	String
	location	String
Messages[]		

getPolicyEditRules – Response Interface

getPurchasingToleranceRules: Input Interface

This table lists the input interface information for the getPurchasingToleranceRules web service operation:

getPurchasingToleranceRules - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	compliance	String					

getPurchasingToleranceRules – Input Interface

getPurchasingToleranceRules: Response Interface

The getPurchasingToleranceRules web service operation uses the ShowPurchasingToleranceRules message as the response. This table lists the response interface information for the getPurchasingToleranceRules web service operation:

ShowPurchasingToleranceRule - Response Interface		
Class	**Field**	**Data Type**
PurchasingToleranceRules []		
	itemId	Integer
	functionType	String
	commodityClass	String
	company	String
	quantityTolerancePercent	BigDecimal
	quantityToleranceUnits	BigDecimal
	unitPriceTolerancePercent	BigDecimal
	unitPriceToleranceAmount	BigDecimal
	extendedAmountTolerancePercent	BigDecimal
	extendedAmountToleranceAmount	BigDecimal
Messages[]		

getPurchasingToleranceRules – Response Interface

getSecurityWorkbench: Input Interface

This table lists the input interface information for the getSecurityWorkbench web service operation:

getSecurityWorkbench - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	compliance	String					

getSecurityWorkbench – Input Interface

getSecurityWorkbench: Response Interface

The getSecurityWorkbench web service operation uses the ShowSecurityWorkbench message as the response. This table lists the response interface information for the getSecurityWorkbench web service operation:

ShowSecurityWorkbench - Response Interface		
Class	**Field**	**Data Type**
SecurityWorkbench []		
	securityType	String
	userId	String
	objectName	String
	runApplication	Boolean
	view	Boolean
	add	Boolean
	change	Boolean
	delete	Boolean
	okSelectAction	Boolean
	copyAction	Boolean
	updateAction	Boolean
Messages[]		

getSecurityWorkbench – Response Interface

Financials

This section lists these interface tables:

- getGLAccount: Input Interface
- getGLAccount: Response Interface
- insertBatchJournalEntry: Input Interface
- insertBatchJournalEntry: Response Interface

getGLAccount: Input Interface

These tables list the input interface information for the getGLAccount web service operation:

getGLAccount - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	description1	String					
	company	String					
	levelOfDetail	String					
	postingEditCode	String					
	currencyCode	String					
	modelOrConsolidationCode	String					
	isTaxable	Boolean					
	taxRateAreaCode	String					
GLAccountKey							
	accountId	String					Y
	accountLongId	String					
	accountAlternate	String					
GLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					
CategoryCodesGLAccount							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					

getGLAccount – Input Interface (1 of 2)

getGLAccount - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					
	categoryCode024	String					
	categoryCode025	String					
	categoryCode026	String					
	categoryCode027	String					
	categoryCode028	String					
	categoryCode029	String					
	categoryCode030	String					
	categoryCode031	String					
	categoryCode032	String					
	categoryCode033	String					
	categoryCode034	String					
	categoryCode035	String					
	categoryCode036	String					
	categoryCode037	String					
	categoryCode038	String					
	categoryCode039	String					
	categoryCode040	String					
	categoryCode041	String					
	categoryCode042	String					
	categoryCode043	String					

getGLAccount – Input Interface (2 of 2)

getGLAccount: Response Interface

The getGLAccount web service operation uses the ShowGLAccount message as the response. These tables list the response interface information for the getGLAccount web service operation:

ShowGLAccount - Response Interface		
Class	**Field**	**Data Type**
	numberRowsReturned	long
GLAccountResult[]		
	description1	String
	company	String
	levelOfDetail	String
	postingEditCode	String
	currencyCode	String
	modelOrConsolidationCode	String
	isTaxable	Boolean
	taxRateAreaCode	String
	objectAccountTarget	String
	subsidiaryTarget	String
	purgeCode	String
	budgetPatternCode	String
	billableCode	String
	unitOfMeasureCode	String
	objectAccountAlternate	String
	subsidiaryAlternate	String
	workersCompensationInsuranceCode	String
	computationMethod	String
	equipmentRateCode	String
	sequenceTypeCode	String
	quantityRollupLevel	String
	costCodeComplete	String
	itemEditCode	String
	postingEditCodeFixedAsset	String
	subledgerProcessingTypeCode	String
GLAccountKey		
	accountId	String
	accountLongId	String
	accountAlternate	String
GLAccount		
	businessUnit	String
	objectAccount	String

getGLAccount – Response Interface (1 of 3)

ShowGLAccount - Response Interface		
Class	**Field**	**Data Type**
	subsidiary	String
GLCostObjects		
	costObjectEditCode1	String
	costObjectEditCode2	String
	costObjectEditCode3	String
	costObjectEditCode4	String
CategoryCodesGLAccount		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
	categoryCode021	String
	categoryCode022	String
	categoryCode023	String
	categoryCode024	String
	categoryCode025	String
	categoryCode026	String
	categoryCode027	String

getGLAccount – Response Interface (2 of 3)

ShowGLAccount - Response Interface		
Class	**Field**	**Data Type**
	categoryCode028	String
	categoryCode029	String
	categoryCode030	String
	categoryCode031	String
	categoryCode032	String
	categoryCode033	String
	categoryCode034	String
	categoryCode035	String
	categoryCode036	String
	categoryCode037	String
	categoryCode038	String
	categoryCode039	String
	categoryCode040	String
	categoryCode041	String
	categoryCode042	String
	categoryCode043	String
Messages[]		

getGLAccount – Response Interface (3 of 3)

insertBatchJournalEntry: Input Interface

These tables list the input interface information for the insertBatchJournalEntry web service operation:

insertBatchJournalEntry - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
InsertBatachJournalEntryStagingFields[]							
	glPostedCode	String					
	batchNumber	Integer					
	batchTypeCode	String					
	timeBatch	Integer					
	company	String					
	periodNumber	Integer					
	century	Integer					
	fiscalYear	Integer					
	fiscalQuarter	String					
	currencyCodeFrom	String					
	rateExchange	Integer					
	rateExchangeHistorical	Integer					
	dateHistorical	Calendar					
	amountDomestic	BigDecimal					
	numberOfUnits	BigDecimal					
	unitOfMeasure	String					
	glOffsetCode	String					
	isReverseOrVoid	String					
	nameExplanationAlpha	String					
	nameExplanationRemark	String					
	reference1	String					
	reference2	String					
	reference3	String					
	checkNumber	String					
	orderTypeCode	String					
	serialNumber	String					
	postCodeBatchRearEnd	String					
	reconciledCode	String					
	isSummarized	String					
	purgeCode	String					

insertBatchJournalEntry – Input Interface (1 of 7)

insertBatchJournalEntry - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	1099Flag	String					
	deleteNotAllowed	String					
	clientFreeForm1	String					
	clientFreeForm2	String					
	postCodeLeaseCostLedger	String					
	billCode	String					
	supplierInvoiceNumber	String					
	categoryWorkOrderCode	String					
	fiscalYearWeekly	Integer					
	fiscalPeriodWeekly	Integer					
	paymentFinal	String					
	sequenceNumber	BigDecimal					
	jobCategory	String					
	jobStep	String					
	businessUnitHome	String					
	divisionOfInterest	Integer					
	leaseNumberAlternate	String					
	leaseTypeAlternateCode	String					
	transactionOriginator	String					
	registrationNumber	Integer					
	paymentId	Integer					
	currencyModeCode	String					
	amountForeign	BigDecimal					
	currencyCodeBase	String					
	amountGrossDomestic	BigDecimal					
	amountGrossForeign	BigDecimal					
	distributionLineNumber	BigDecimal					
	receiptNumber	String					
	postedBurdeningCode	String					
	businessUnitHomeAlternate	String					
	employeePoolGroupingCode	String					

insertBatchJournalEntry – Input Interface (2 of 7)

insertBatchJournalEntry - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	jobPoolGroupingCode	String					
	costingActivityBasedCode	String					
	itemId	Integer					
EDITransaction							
	ediUserId	String					Y
	ediRecordType	String					
	ediRecordSequence	Integer					
	ediTransactionNumber	String					Y
	editDocumentType	String					
	ediLineNumber	BigDecimal					Y
	ediTransactionSetNumber	String					
	ediTranslationFormat	String					
	ediTransmissionDate	Calendar					
	ediSendReceiveIndicator	String					
	ediDetailLinesProcessed	Integer					
	ediSuccessfullyProcessed	String					
	ediTransactionAction	String					
	ediTransactionType	String					
	ediBatchNumber	String					Y
	ediBatchFileCreateGLRecord	String					
	ediUserId	Integer					
GeneralLedgerKey							
	documentCompany	String					
	documentTypeCode	String					
	documentNumber	Integer					
	dateAccounting	Calendar					
	documentLineNumber	Integer					
	lineExtensionCode	String					
	ledgerTypeCode	String					
	documentPayItem	String					

insertBatchJournalEntry – Input Interface (3 of 7)

insertBatchJournalEntry - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
GLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					
GLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					
Subledger							
	subledger	String					
	subledgerTypeCode	String					
Entity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
OriginalOrderKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentSuffix	String					
	documentCompany	String					
PurchaseOrderKey							
	documentCompany	String					
	documentNumber	String					
	documentTypeCode	String					
	documentSuffix	String					
	documentLineNumber	BigDecimal					
GLAlternatePostCodes							
	postCodeAlternate0	String					
	postCodeAlternate1	String					
	postCodeAlternate2	String					
	postCodeAlternate3	String					
	postCodeAlternate4	String					

insertBatchJournalEntry – Input Interface (4 of 7)

insertBatchJournalEntry - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	postCodeAlternate5	String					
	postCodeAlternate6	String					
	postCodeAlternate7	String					
	postCodeAlternate8	String					
	postCodeAlternate9	String					
	postCodeAlternateT	String					
	postCodeAlternateU	String					
	postCodeAlternateV	String					
	postCodeAlternateW	String					
	postCodeAlternateX	String					
	postCodeAlternateZ	String					
Dates							
	dateCheck	Calendar					
	dateCheckCleared	Calendar					
	dateServiceTax	Calendar					
	dateInvoice	Calendar					
	dateBatch	Calendar					
	dateBatchSystem	Calendar					
	dateGLMonth	Integer					
	dateGLDay	Integer					
	dateGLYear	Integer					
	dateGLCentury	Integer					
	dateBatchMonth	Integer					
	dateBatchDay	Integer					
	dateBatchYear	Integer					
	dateBatchCentury	Integer					
	dateBatchSystemMonth	Integer					
	dateBatchSystemDay	Integer					
	dateBatchSystemYear	Integer					
	dateBatchSystemCentury	Integer					
	dateCheckMonth	Integer					
	dateCheckDay	Integer					

insertBatchJournalEntry – Input Interface (5 of 7)

insertBatchJournalEntry - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	dateCheckYear	Integer					
	dateCheckCentury	Integer					
	dateServiceTaxMonth	Integer					
	dateServiceTaxDay	Integer					
	dateServiceTaxYear	Integer					
	dateServiceTaxCentury	Integer					
	dateHistoricalMonth	Integer					
	dateHistoricalDay	Integer					
	dateHistoricalYear	Integer					
	dateHistoricalCentury	Integer					
	dateCheckClearedMonth	Integer					
	dateCheckClearedDay	Integer					
	dateCheckClearedYear	Integer					
	dateCheckClearedCentury	Integer					
	dateInvoiceMonth	Integer					
	dateInvoiceDay	Integer					
	dateInvoiceYear	Integer					
	dateInvoiceCentury	Integer					
GLCostObjects							
	costObjectCode1	String					
	costObjectCode2	String					
	costObjectCode3	String					
	costObjectCode4	String					
	costObjectTypeCode1	String					
	costObjectTypeCode2	String					
	costObjectTypeCode3	String					
	costObjectTypeCode4	String					
GLPostCodes							
	postingCode1	String					
	postingCode2	String					
	postingCode3	String					
	postingCode4	String					

insertBatchJournalEntry – Input Interface (6 of 7)

insertBatchJournalEntry - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	postingCode5	String					
	postingCode6	String					
	postingCode7	String					
	postingCode8	String					
	postingCode9	String					
	postingCode10	String					
Tax							
	taxExplanationCode	String					
	taxRateAreaCode	String					
	itemIdTax	Integer					
	amountTaxDomestic	BigDecimal					
	amountTaxForeign	BigDecimal					
	trackTaxes	String					

insertBatchJournalEntry – Input Interface (7 of 7)

insertBatchJournalEntry: Response Interface

The insertBatchJournalEntry web service operation uses the ConfirmInsertBatchJournalEntry message as the response. This table lists the response interface information for the insertBatchJournalEntry web service operation:

ConfirmInsertBatchJournalEntry - Response Interface		
Class	Field	Data Type
	numberRecordsInserted	long
Messages[]		

insertBatchJournalEntry – Response Interface

Fixed Assets

This section lists these interface tables:

- getFixedAsset: Input Interface
- getFixedAsset: Response Interface

getFixedAsset: Input Interface

These tables list the input interface information for the getFixedAsset web service operation:

getFixedAsset - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	company	String					
	businessUnitOriginal	String					
	subledgerInactiveCode	String					
	purchaseOptionCode	String					
	statusEquipmentCode	String					
	stateCode	String					
Asset							
	assetId	Integer					**Y**
	unitNumber	String					
	assetIdParent	Integer					
	serialNumber	String					
	sequenceNumber	Integer					
Entity - taxEntity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - lessorEntity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Dates							
	dateAcquired	Calendar					
	dateDisposal	Calendar					
	dateEffective	Calendar					
	dateExpiration	Calendar					
Insurance							
	insurancePolicyNumber	String					
	insuranceCompany	String					
	insurancePolicyRenewalMonth	Integer					
	insurancePremium	BigDecimal					
	insuranceValue	BigDecimal					

getFixedAsset – Input Interface (1 of 2)

getFixedAsset - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
CategoryCodes							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
	categoryCode011	String					
	categoryCode012	String					
	categoryCode013	String					
	categoryCode014	String					
	categoryCode015	String					
	categoryCode016	String					
	categoryCode017	String					
	categoryCode018	String					
	categoryCode019	String					
	categoryCode020	String					
	categoryCode021	String					
	categoryCode022	String					
	categoryCode023	String					

getFixedAsset – Input Interface (2 of 2)

getFixedAsset: Response Interface

The getFixedAsset web service operation uses the ShowFixedAsset message as the response. These tables list the response interface information for the getFixedAsset web service operation:

ShowFixedAsset - Response Interface		
Class	**Field**	**Data Type**
FixedAsset[]		
	company	String
	businessUnitOriginal	String
	subledgerInactiveCode	String
	purchaseOptionCode	String
	remark1	String
	remark2	String
	businessUnitLocation	String
	itemParent	Integer
	item	String
	standardFuelConsumption	BigDecimal
	description1	String
	description2	String
	description3	String
	descriptionCompressed	String
	statusEquipmentCode	String
	newOrUsedOnAcquisitionCode	String
	methodOfFinanceCode	String
	stateCode	String
Entity - taxEntity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
Entity - lessorEntity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
Asset		
	assetId	Integer
	unitNumber	String
	assetIdParent	Integer
	serialNumber	String
	sequenceNumber	Integer

getFixedAsset – Response Interface (1 of 3)

ShowFixedAsset - Response Interface		
Class	**Field**	**Data Type**
GLAccount - glAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
GLAccount - depreciationGLAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
GLAccount - revenueGLAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
Dates		
	dateAcquired	Calendar
	dateDisposal	Calendar
	dateEffective	Calendar
	dateExpiration	Calendar
Finance		
	amountEstimatedSalvageValue	BigDecimal
	amountReplacementCost	BigDecimal
	amountLastYearsReplacement	BigDecimal
	amountInvestmentTaxCreditYTD	BigDecimal
	amountInvestmentTaxCreditPriorYear	BigDecimal
	amountPurchaseOptionPrice	BigDecimal
	amountPurchaseOptionPercentage	BigDecimal
	amountPurchaseOptionPriceMax	BigDecimal
	amountPayment	BigDecimal
Quantity		
	quantityOnHand	BigDecimal
	quantityActual	BigDecimal
Insurance		
	insurancePolicyNumber	String
	insuranceCompany	String
	insurancePolicyRenewalMonth	Integer

getFixedAsset – Response Interface (2 of 3)

ShowFixedAsset - Response Interface		
Class	**Field**	**Data Type**
	insurancePremium	BigDecimal
	insuranceValue	BigDecimal
CategoryCodes		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
	categoryCode011	String
	categoryCode012	String
	categoryCode013	String
	categoryCode014	String
	categoryCode015	String
	categoryCode016	String
	categoryCode017	String
	categoryCode018	String
	categoryCode019	String
	categoryCode020	String
	categoryCode021	String
	categoryCode022	String
	categoryCode023	String

getFixedAsset – Response Interface (3 of 3)

Foundation Environment

This section lists these interface tables:

- getUserDefinedCode: Input Interface
- getUserDefinedCode: Response Interface

getUserDefinedCode: Input Interface

This table lists the input interface information for the getUserDefinedCode web service operation:

getUserDefinedCode - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	userDefinedCode	String					Y
	description1	String					
	description2	String					
	specialHandlingCode	String					

getUserDefinedCode – Input Interface

getUserDefinedCode: Response Interface

The getUserDefinedCode web service operation uses the ShowUserDefinedCode message as the response. This table lists the response interface information for the getUserDefinedCode web service operation:

ShowUserDefinedCode - Response Interface		
Class	Field	Data Type
showUserDefinedCode[]		
	productCode	String
	userDefinedCodeTable	String
	userDefinedCode	String
	description1	String
	description2	String
	specialHandlingCode	String
	hardCoded	Boolean
messages[]		
	messages	String

getUserDefinedCode – Response Interface

Inventory

This section lists these interface tables:

- processSupplierCatalogPrice: Input Interface
- processSupplierCatalogPrice: Response Interface
- processInventoryItem: Input Interface
- processInventoryItem: Response Interface
- getBranchPlantItem: Input Interface
- getBranchPlantItem: Response Interface
- getSupplierCatalogPrice: Input Interface
- getSupplierCatalogPrice: Response Interface
- getItemAvailability: Input Interface
- getItemAvailability: Response Interface
- insertInventoryItemStaging: Input Interface
- insertInventoryItemStaging: Response Interface

processSupplierCatalogPrice: Input Interface

This table lists the input interface information for the processSupplierCatalogPrice web service operation:

processSupplierCatalogPrice - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
processing							
	actionType	String	Y	Y			
	catalogName	String	Y	Y			
item							
	itemId	Integer	Y	Y			
	itemProduct	String					
	itemCatalog	String					
	itemSupplier	String					
	itemFreeForm	String					
supplier							
	entityId	Integer	Y	Y			
	entityLongId	String					
	entityTaxId	String					
	businessUnit	String					
	priceUnit	BigDecimal					
	currencyCode	String					
	quantityActual	BigDecimal					
	unitOfMeasureCode	String					
	dateEffectiveEnd	Calendar					
	dateEffectiveStart	Calendar					

processSupplierCatalogPrice – Input Interface

processSupplierCatalogPrice: Response Interface

The processSupplierCatalogPrice web service operation uses the ConfirmProcessSupplierCatalogPrice message as the response. This table lists the response interface information for the processSupplierCatalogPrice web service operation:

ConfirmProcessSupplierCatalogPrice - Response Interface		
Class	**Field**	**Data Type**
	businessUnit	String
supplier		Entity
	entityId	Integer
	entityLongId	String
	entityTaxId	String
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemSupplier	String
	catalogName	String
	currencyCode	String
	priceUnit	BigDecimal
	unitOfMeasureCode	String
	quantityActual	BigDecimal
	dateEffectiveStart	Calendar
	dateEffectiveEnd	Calendar
messages[]		
	messages	E1MessageList

processSupplierCatalogPrice – Response Interface

processInventoryItem: Input Interface

This table lists the input interface information for the processInventoryItem web service operation:

processInventoryItem - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
processing							
	actionTypeCode	String	Y	Y			
	version	String					
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemSupplier	String					
	itemFreeForm	String					
itemDimensions							
	unitOfMeasureCodePrimary	String					
	unitOfMeasureCodeWeight	String					
	unitOfMeasureCodeVolume	String					
	description1	String					
	description2	String					
	stockingTypeCode	String					
	lineTypeCode	String					
	glClassCode	String					
	serialNumberFlag	String					
	lotProcessCode	String					
	lotStatusCode	String					
	daysShelfLife	Integer					

processInventoryItem – Input Interface

processInventoryItem: Response Interface

The processInventoryItem web service operation uses the ConfirmProcessInventoryItem message as the response. These tables list the response interface information for the processInventoryItem web service operation:

ConfirmProcessInventoryItem - Response Interface		
Class	**Field**	**Data Type**
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemSupplier	String
	descriptionLine1	String
	descriptionLine2	String
	searchText	String
	commodityCode	String
	productGroupFrom	String
	dispatchGroup	String
	pricingCategory	String
	repriceBasketPriceCat	String
	orderRepriceCategory	String
	buyer	Integer
itemDimensions		
	volumeCubicDimensions	BigDecimal
	unitOfMeasurePrimary	String
	unitOfMeasureSecondary	String
	unitOfMeasurePurchas	String
	unitOfMeasurePricing	String
	unitOfMeasureShipping	String
	unitOfMeasureProduction	String
	unitOfMeasureComponent	String
	unitOfMeasureWeight	String
	unitOfMeasureVolume	String
	unitOfMeasureStocking	String
	glCategory	String
	priceLevel	String
	levelPurchasingPrice	String
	costLevel	String
	checkAvailabilityYN	String

processInventoryItem – Response Interface (1 of 3)

ConfirmProcessInventoryItem - Response Interface		
Class	**Field**	**Data Type**
	bulkPackedFlag	String
	layerCodeSource	String
	stockingType	String
	lineType	String
	unitofMeasureVolumeorWel	String
	componentType	String
	contractItem	String
	backordersAllowedYN	String
	itemFlashMessage	String
	printMessage1	String
	abcCode1SalesInv	String
	abcCode2MarginInv	String
	abcCode3InvestInv	String
	abcCodeOverrideIndica	String
	typeWarranty	String
	commissionCategory	String
	serialNumberRequired	String
	lotStatusCode	String
	shelfLifeDays	Integer
	fifoProcessing	String
	addressNumberPlanner	Integer
	percentMargin	BigDecimal
	marginMaintenancePer	BigDecimal
	orderWithYN	String
	expediteDamperDays	Integer
	deferDamperDays	Integer
	safetyLeadtime	Integer
	makeBuyCode	String
	coproductsByproducts	String
	lowLevelCode	Integer
	allocateByLot	String
	commitmentSpecificDays	Integer

processInventoryItem – Response Interface (2 of 3)

ConfirmProcessInventoryItem - Response Interface		
Class	**Field**	**Data Type**
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
	searchTextCompressed	String
	segment1	String
	segment2	String
	segment3	String
	segment4	String
	segment5	String
	segment6	String
	segment7	String
	segment8	String
	segment9	String
	segment10	String
	dualUnitOfMeasureItem	String
	dualPickingProcessOption	String
	dualTolerance	BigDecimal
	purchasingEffectiveDays	Integer
	sellByDefaultDays	Integer
	userLotDate1DefaultDays	Integer
	userLotDate2DefaultDays	Integer
	userLotDate3DefaultDays	Integer
	userLotDate4DefaultDays	Integer
	userLotDate5DefaultDays	Integer
	crossDockFlagYN	String
	lotAuditFlag	String
	vendorManagedInventory	String
messages[]		
	messages	String

processInventoryItem – Response Interface (3 of 3)

getBranchPlantItem: Input Interface

These tables list the input interface information for the getBranchPlantItem web service operation:

getBranchPlantItem Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	branchPlant	String					
Item							
	itemId	Integer					
	itemCatalog	String					
	itemProduct	String					
	description1	String					
	description2	String					
	stockingTypeCode	String					
	lineTypeCode	String					
SalesCategoryCodes							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					
PurchasingCategoryCodes							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
	categoryCode006	String					
	categoryCode007	String					
	categoryCode008	String					
	categoryCode009	String					
	categoryCode010	String					

getBranchPlantItem – Input Interface (1 of 2)

getBranchPlantItem Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Buyer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Planner							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

getBranchPlantItem – Input Interface (2 of 2)

getBranchPlantItem: Response Interface

The getBranchPlantItem web service operation uses the ShowBranchPlantItem message as the response. These tables list the response interface information for the getBranchPlantItem web service operation:

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	numberRowsReturned	long
Item		
	itemId	Integer
	itemCatalog	String
	itemProduct	String
ItemBranch		
	branchPlant	String
	stockingTypeCode	String
	productFamilyCode	String
	primaryUnitOfMeasureCode	String
	description1	String
	description2	String
	ABCCodeInvestment	String
	ABCCodeMargin	String
	ABCCodeSales	String
	activityBasedCostingOverride	String
	isBackOrderAllowed	Boolean
	orderRepriceCategoryCode	String
	checkAvailability	Boolean
	countyCode	String
	glClassCode	String
	printMessageCode	String
	constraintsCode	String
	taxableCodePurchasing	String
	taxableCodeSales	String
	commitmentMethodCode	String
	repriseBasketPriceCode	String
	lineTypeCode	String
	processCrossDockingFlag	String
	componentTypeCode	String
	warrantyType	String

getBranchPlantItem – Response Interface (1 of 7)

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	commissionCategory	String
	FIFOProcessing	String
	percentMarginMaintenance	BigDecimal
	percentMargin	BigDecimal
	isOrderWith	Boolean
	priorityAlertLevel1	BigDecimal
	priorityAlertLevel2	BigDecimal
	isMixedDatesOrLots	Boolean
	dateLIFO	Calendar
	quantityUnitsBatch	BigDecimal
	itemDisplayCode	String
	applyFrieghtCode	String
	serviceLevel	BigDecimal
	quantityOrderMultiplesWorkOrder	Integer
	entityIdSupplier	Integer
	entityIdPlanner	Integer
	entityIdBuyer	Integer
	entityIdCarrier	Integer
	entityIdPreferredCarrier	Integer
ItemMaster		
	commodityClassCode	String
	dispatchGroup	String
	itemPriceGroupCode	String
	cycleCountCategory	String
	commodityClassCodeShipping	String
	shippingConditionsCode	String
	itemFlashMessageCode	String
	roundToWholeNumberCode	String
	unitOfMeasureConversionCodeStandard	String
PurchasingCategoryCodes		
	categoryCode001	String

getBranchPlantItem – Response Interface (2 of 7)

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
SalesCategoryCodes		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
Potency		
	controlPotency	Boolean
	potencyStandard	BigDecimal
	potencyFrom	BigDecimal
	potencyTo	BigDecimal
Grade		
	controlGrade	Boolean
	gradeCodeStandard	String
	gradeCodeFrom	String
	gradeCodeTo	String
Quantities		
	quantityReorder	BigDecimal

getBranchPlantItem – Response Interface (3 of 7)

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	quantityReorderMaximum	BigDecimal
	quantityReorderMinimum	BigDecimal
	reorderPoint	BigDecimal
	quantityOrderMultiples	BigDecimal
	unitsPerContainer	Integer
	quantitySafetyStock	BigDecimal
UserReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
LotProcessing		
	serialNumberRequiredCode	String
	lotStatusCode	String
	lotProcessTypeCode	String
	commitmentDateMethodCode	String
	lotExpirationDateMethodCode	String
	daysShelfLife	Integer
	daysBestBeforeResult	Integer
	daysSellByDefault	Integer
	daysEffectiveManufacturing	Integer
	daysEffectivePurchasing	Integer
	daysUserLotDateDefault1	Integer
	daysUserLotDateDefault2	Integer
	daysUserLotDateDefault3	Integer
	daysUserLotDateDefault4	Integer
	daysUserLotDateDefault5	Integer
	allowShippingOfHeldLots	String
	lotStatusCodeExpanded	String
	isLotNumberPreAssigned	String
	isLotNumberAssigmentRestricted	String

getBranchPlantItem – Response Interface (4 of 7)

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	lotFormatSpecial	String
	lotAuditFlag	String
PlantManufacturing		
	orderPolicyCode	String
	orderPolicyNumber	Integer
	planningCode	String
	planningFenceRuleCode	String
	daysTimeFencePlanning	Integer
	daysTimeFenceFreeze	Integer
	daysTimeFenceMessage	Integer
	daysTimeFenceManufacturing	Integer
	daysShipmentLeadtimeFence	Integer
	daysExpiditeDamper	Integer
	daysDeferDamper	Integer
	daysCommitmentSpecific	Integer
	hoursStandardSetupLabor	BigDecimal
	hoursStandardQueue	BigDecimal
	hoursStandardMove	BigDecimal
	ECONumber	String
	ECOReasonCode	String
	dateECO	Calendar
	issueAndReceiptCode	String
	hoursReplenishment	BigDecimal
	isActiveIngredient	Boolean
	issueTypeCode	String
	timeBasisCode	String
	itemRevisionLevel	String
	shrinkFactorNumber	BigDecimal
	shrinkFactorMethodCode	String
	leadtimeLevel	Integer
	leadtimeManufacturing	Integer
	leadtimeCumulative	Integer

getBranchPlantItem – Response Interface (5 of 7)

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	leadtimePerUnit	BigDecimal
	leadtimeFixedOrVariableCode	String
	leadtimeSafety	Integer
	leadtimeConversionFactor	BigDecimal
	quantityManufacturingLeadtime	BigDecimal
	quantityAccountingCost	BigDecimal
	makeOrBuyCode	String
	lowLevelCode	Integer
AdditionalUnitsOfMeasure		
	unitOfMeasureVolumeorWeight	String
	dualUnitOfMeasureItem	String
	unitOfMeasureSecondary	String
	unitOfMeasurePurchasing	String
	unitOfMeasurePricing	String
	unitOfMeasureShipping	String
	unitOfMeasureProduction	String
	unitOfMeasureWeight	String
	unitOfMeasureVolume	String
	unitOfMeasureStocking	String
	unitOfMeasureUPC	String
	unitOfMeasureAggregateUPC	String
	unitOfMeasureSCCPI0	String
	unitOfMeasureSCCPI1	String
	unitOfMeasureSCCPI2	String
	unitOfMeasureSCCPI3	String
	unitOfMeasureSCCPI4	String
	unitOfMeasureSCCPI5	String
	unitOfMeasureSCCPI6	String
	unitOfMeasureSCCPI7	String
	unitOfMeasureSCCPI8	String
	operationalThresholdUOM	String
	cumulativeThresholdUOM	String

getBranchPlantItem – Response Interface (6 of 7)

ShowBranchPlantItem - Response Interface		
Class	**Field**	**Data Type**
	APSPlanningUOM	String
Buyer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
Planner		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
Supplier		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
messages[]		
	messages	String

getBranchPlantItem – Response Interface (7 of 7)

getSupplierCatalogPrice: Input Interface

This table lists the input interface information for the getSupplierCatalogPrice web service operation:

getSupplierCatalogPrice - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	businessUnit	String					
Supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Item GroupSupplier							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemSupplier	String					
	itemFreeForm	String					
	catalogName	String					
	agreementId	String					
	agreementSupplement	Integer					
	currencyCode	String					
	unitOfMeasureCode	String					
	quantityActual	BigDecimal					
	dateEffectiveStart	Calendar					
	dateEffectiveEnd	Calendar					
	priceUnit	BigDecimal					
	documentNumber	Integer					
	documentType	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					

getSupplierCatalogPrice – Input Interface

getSupplierCatalogPrice: Response Interface

The getSupplierCatalogPrice web service operation uses the ShowSupplierCatalogPrice message as the response. :This table lists the response interface information for the getSupplierCatalogPrice web service operation:

ShowSupplierCatalogPrice - Response Interface		
Class	**Field**	**Data Type**
	numberRowsReturned	long
	businessUnit	String
	entityIdSupplier	Integer
Item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	catalogName	String
	agreementId	String
	agreementSupplement	Integer
	currencyCode	String
	unitOfMeasureCode	String
	priceUnit	BigDecimal
	quantityActual	BigDecimal
	dateEffectiveStart	Calendar
	dateEffectiveEnd	Calendar
	documentNumber	Integer
	documentType	String
	documentCompany	String
	documentLineNumber	BigDecimal

getSupplierCatalogPrice – Response Interface

getItemAvailability: Input Interface

This table lists the input interface information for the getItemAvailability web service operation:

getItemAvailability - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	businessUnit	String				Y	
	transactionUnitOfMeasure	String				Y	

getItemAvailability – Input Interface

getItemAvailability: Response Interface

The getItemAvailability web service operation uses the ShowItemAvailability message as the response. This table lists the response interface information for the getItemAvailability web service operation:

ShowItemAvailability - Response Interface		
Class	**Field**	**Data Type**
	businessUnit	String
	transactionUnitOfMeasure	String
	quantityAvailable	BigDecimal
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
messages[]		
	messages	E1MessageList

getItemAvailability – Response Interface

insertInventoryItemStaging: Input Interface

This table lists the input interface information for the insertInventoryItemStaging web service operation:

insertInventoryItemStaging - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
insertInventoryItemStagingFields[ ]							
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
itemDimensions							
	unitOfMeasureCodePrimary	String					
	unitOfMeasureCodeSecondary	String					
	unitOfMeasureCodePurchasing	String					
	unitOfMeasureCodePricing	String					
	unitOfMeasureCodeWeight	String					
	unitOfMeasureCodeVolume	String					
	unitOfMeasureCodeUPCNumber	String					
processing							
	actionTypeCode	String					
	description1	String					
	description2	String					
	stockingTypeCode	String					
	itemGrade	String					
	lineTypeCode	String					
	glClassCode	String					
	serialNumberFlag	String					
	lotProcessCode	String					
	lotStatusCode	String					
	daysShelfLife	Integer					
	upcNumber	String					

insertInventoryItemStaging – Input Interface

insertInventoryItemStaging: Response Interface

The insertInventoryItemStaging web service operation uses the ConfirmInsertInvetoryItemStaging message as the response. This table lists the response interface information for the insertInventoryItemStaging web service operation:

ConfirmInsertInventoryItemStaging - Response Interface		
Class	**Field**	**Data Type**
	numRowsInserted	long

insertInventoryItemStaging – Response Interface

Procurement

This section lists these interface tables:

- processPurchaseOrder: Input Interface
- processPurchaseOrder: Response Interface
- processPurchaseOrderAcknowledge: Input Interface
- processPurchaseOrderAcknowledge: Response Interface
- getPurchaseOrder: Input Interface
- getPurchaseOrder: Response Interface

processPurchaseOrder: Input Interface

These tables list the input interface information for the processPurchaseOrder web service operation:

processPurchaseOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
Header							
	businessUnit	String	Y				
	orderedBy	String					
	orderTakenBy	String					
	holdOrderCode	String					
	supplierSO	String					
	reference	String					
	remark	String					
	description1	String					
	printMessageCode	String					
	sendMethodCode	String					
	percentRetainage	BigDecimal					
	evaluatedReceiptsSettlement	String					
	supplierPriceGroupCode	String					
	adjustmentScheduleCode	String					
	changeOrderNumber	Integer					
	RMAId	Integer					
	RMAType	String					
	currencyCodeFrom	String					
	rateExchangeOverride	Integer					
	currencyCodeTo	String					
	paymentTermsCode	String					
	triangulationRateToCurrency	Integer					
	triangulationRateFromCurrency	Integer					
	isTextAssociated	Boolean					
	approvalRouteCode	String					
	approvalRoutePO	String					
	printAIADocument	String					
Processing							
	processingVersion	String					
	actionType	String	Y	Y	Y		

processPurchaseOrder – Input Interface (1 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
PurchaseOrderKey							
	documentNumber	Integer		Y	Y		
	documentTypeCode	String		Y	Y		
	documentCompany	String		Y	Y		
Buyer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Dates							
	dateOrdered	Calendar					
	dateRequested	Calendar					
	datePromisedDelivery	Calendar					
	dateCancel	Calendar					
Supplier			Y				
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Address							
	mailingName	String					
	addressLine1	String					
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
ShipTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

processPurchaseOrder – Input Interface (2 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Address							
	mailingName	String					
	addressLine1	String					
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
Tax							
	taxRateAreaCode	String					
	taxExplanationCode	String					
	taxExemptCertificate	String					
Delivery							
Carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	deliveryInstruction1	String					
	deliveryInstruction2	String					
	freightHandlingCode	String					
	landedCostRuleCode	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					

processPurchaseOrder – Input Interface (3 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Detail[]							
PurchaseOrderLineKey							
	documentLineNumber	BigDecimal		Y	Y		
	documentSuffix	String					
	actionType	String	Y	Y	Y		
	transactionOriginator	String					
	businessUnit	String					
	supplierSO	String					
	reference	String					
	reference1	String					
	agreementId	Integer					
	agreementSupplement	Integer					
	transferDirectShip	String					
	printMessageCode	String					
	promotionId	String					
	discountFactor	BigDecimal					
	adjustmentScheduleCode	String					
	adjustmentRevisionLevel	Integer					
	freezeWorkOrder	String					
	relievePOBlanketOrder	String					
	messageCode	String					
	statusCodeMaterial	String					
	costObjectCode1	String					
	costObjectCode2	String					
	costObjectCode3	String					
	costObjectCode4	String					
	costObjectTypeCode1	String					
	costObjecTypeCode2	String					
	costObjectTypeCode3	String					
	costObjectTypeCode4	String					
	shipmentId	Integer					
	supplierPriceGroupCode	String					

processPurchaseOrder – Input Interface (4 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	priceLevel	String					
	RMAId	Integer					
	RMAType	String					
	isAPSActive	Boolean					
	matrixControlLine	BigDecimal					
	UNSPSCCode	String					
	commodityClassCode	String					
	statusCodeNext	String					
	statusCodeLast	String					
	quantityOrdered	BigDecimal	Y				
	quantityOrderedSecondary	BigDecimal					
	unitOfMeasureCodeSecondary	String					
	unitOfMeasureCodeTransaction	String					
	unitOfMeasureCodeWeight	String					
	unitOfMeasureCodeVolume	String					
	unitVolume	BigDecimal					
	unitWeight	BigDecimal					
	costExtended	BigDecimal					
	costUnitPurchasing	BigDecimal					
	unitOfMeasureCodePurchasing	String					
Buyer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
ShipTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Dates							
	dateRequested	Calendar					
	datePromisedDelivery	Calendar					
	dateAccounting	Calendar					

processPurchaseOrder – Input Interface (5 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	dateCancel	Calendar					
	datePromisedShip	Calendar					
	dateEffectiveLot	Calendar					
Product							
Item							
	itemId	Integer	Y				
	itemProduct	String					
	itemCatalog	String					
	itemSupplier	String					
	itemFreeForm	String					
	description1	String					
	description2	String					
	itemUPCEAN	String					
	locationPrimary	String					
	lotNumber	String					
	lineTypeCode	String					
Financial							
GLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					
GLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					
	assetId	String					
	subledger	String					
	subledgerTypeCode	String					
	glClassCode	String					
Tax							
	taxableCode	String					
	taxRateAreaCode	String					

processPurchaseOrder – Input Interface (6 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	taxExplanationCode	String					
Delivery							
Carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	modeOfTransportCode	String					
	landedCostRuleCode	String					
	freightHandlingCode	String					
OriginalOrderLineKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	documentSuffix	String					
RelatedOrderLineKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	documentSuffix	String					
Project							
	projectNumber	Integer					
	taskOrderNumber	Integer					
	taskOrderType	String					
CategoryCodesPurchasing							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					

processPurchaseOrder – Input Interface (7 of 8)

processPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
CategoryCodesSales							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					

processPurchaseOrder – Input Interface (8 of 8)

processPurchaseOrder: Response Interface

The processPurchaseOrder web service operation uses the ConfirmPurchaseOrder message as the response. This table lists the response interface information for the processPurchaseOrder web service operation:

ConfirmPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
Header		
	DocumentNumber	Integer
	DocumentCompany	String
	DocumentTypeCode	String
	Document Suffix	String
	Hold Code	String
	Reference	String
	Reference2	String
Financial		
	Domestic Order Total	BigDecimal
	Foreign Order Total	BigDecimal
Detail[]		
	Purchase Order Line Number	BigDecimal
	Business Unit	String
Quantity		
	Quantity Ordered	BigDecimal
	Transaction Unit Of Measure	String
Product		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemSupplier	String
	itemFreeForm	String
Financial		
	Purchasing Unit Of Measure	String
	Foreign Purchasing Unit Cost	BigDecimal
	Domestic Purchasing Unit Cost	BigDecimal
	Foreign Extended Cost	BigDecimal
	Domestic Extended Cost	BigDecimal
messages[]		
	messages	E1MessageList

processPurchaseOrder – Response Interface

processPurchaseOrderAcknowledge: Input Interface

These tables list the input interface information for the processPurchaseOrderAcknowledge web service operation:

processPurchaseOrderAcknowledge - Input Interface			Required				Key
Class	**Fields**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
Processing							
	processingVersion	String					
Header							
	statusOrderCode	String		Y	Y		
	orderTakenBy	String					
	attachmentText	String					
	freightHandlingCode	String					
PurchaseOrderKey							
	documentNumber	Integer		Y	Y		
	documentCompany	String		Y	Y		
	documentTypeCode	String		Y	Y		
Financial							
	rateExchangeOverride	BigDecimal					
	paymentTermsCode	String					
Tax							
	taxRateAreaCode	String					
	taxExplanationCode	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					
Detail							
	statusOrderCode	String		Y	Y		
	reference1	String					
	reference2	String					
	quantityShippable	BigDecimal					
	unitOfMeasureCodeTransaction	String					
	landedCostRuleCode	String					
	branchPlant	String					

processPurchaseOrderAcknowledge – Input Interface (1 of 3)

processPurchaseOrderAcknowledge - Input Interface			Required				Key
Class	Fields	Data Type	A	C	D/C	I	(Y/N)
PurchaseOrderLineKey							
	documentLineNumber	BigDecimal		Y	Y		
	documentSuffix	String					
ItemGroupSupplier							
	itemId	Integer		Y	Y		
	itemProduct	String					
	itemCatalog	String					
	itemFreeForm	String					
	itemSupplier	String					
Supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Dates							
	dateRequested	Calendar					
	datePromisedDelivery	Calendar					
Financial							
	unitOfMeasureCodePurchasing	String					
	costUnit	BigDecimal					
	costExtended	BigDecimal					
Tax							
	taxableCode	String					
	taxRateAreaCode	String					
	taxExplanationCode	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					

processPurchaseOrderAcknowledge – Input Interface (2 of 3)

processPurchaseOrderAcknowledge - Input Interface			Required				Key
Class	Fields	Data Type	A	C	D/C	I	(Y/N)
	userReservedNumber	Integer					
	userReservedReference	String					

processPurchaseOrderAcknowledge – Input Interface (3 of 3)

processPurchaseOrderAcknowledge: Response Interface

The processPurchaseOrderAcknowledge web service operation uses the ConfirmPurchaseOrderAcknowledge message as the response. This table lists the response interface information for the processPurchaseOrderAcknowledge web service operation:

ConfirmPurchaseOrderAcknowledge - Response Interface		
Class	**Field**	**Data Type**
Header		
PurchaseOrderKey		
	DocumentNumber	Integer
	DocumentCompany	String
	DocumentTypeCode	String
Detail		
PurchaseOrderLineKey		
	DocumentLineNumber	BigDecimal
	DocumentSuffix	String
	Item	Integer
	quantityShippable	BigDecimal
	costUnit	BigDecimal
	CostExtended	BigDecimal
messages[]		
	messages	E1MessageList

processPurchaseOrderAcknowledge – Response Interface

getPurchaseOrder: Input Interface

These tables list the input interface information for the getPurchaseOrder web service operation:

getPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Header							
PurchaseOrderKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
	businessUnit	String					
	deliveryInstruction1	String					
	deliveryInstruction2	String					
	paymentTermsCode	String					
	taXExplanationCode	String					
	taxRateAreaCode	String					
	holdCode	String					
	freightHandlingCode	String					
	zoneNumber	Integer					
	currencyCodeFrom	String					
	rateExchangeOverride	BigDecmial					
	orderedBy	String					
	orderTakenBy	String					
Dates							
	dateTransaction	Calendar					
	dateCancel	Calendar					
	datePromisedShip	Calendar					
Buyer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
MarkFor							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

getPurchaseOrder – Input Interface (1 of 4)

getPurchaseOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					
Detail							
	PurchaseOrderLineKey						
	documentLineNumber	BigDecimal					
	documentSuffix	String					
	businessUnit	String					
	reference1	String					
	reference2	String					
	statusCodeNext	String					
	statusCodeLast	String					
	priceUnitDomestic	BigDecimal					
	taxExplanationCode	String					
	taxRateAreaCode	String					
	modeOfTransportCode	String					
	unitOfMeasureCodePurchasing	String					
	rateExchangeOverride	BigDecmial					
	costUnitPurchasingDomestic	BigDecmial					
	priceExtendedDomestic	BigDecmial					
	priceUnitForeign	BigDecmial					
	costUnitPurchasingForeign	BigDecmial					
	priceExtendedForeign	BigDecmial					
	amountTaxForeign	BigDecmial					
Item Supplier Group							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					

getPurchaseOrder – Input Interface (2 of 4)

getPurchaseOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	itemCustomer	String					
	itemFreeForm	String					
OriginalOrderKey							
	documentNumber	Integer					
	documentCompany	String					
	documentTypeCode	String					
OriginalOrderLineKey							
	documentLineNumber	BigDecimal					
	documentSuffix	String					
RelatedOrderKey							
	documentNumber	Integer					
	documentCompany	String					
	documentTypeCode	String					
RelatedOrderLineKey							
	documentLineNumber	BigDecimal					
	documentSuffix	String					
Supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
ShipTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Buyer							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Dates							
	dateRequested	Calendar					
	dateScheduledPick	Calendar					

getPurchaseOrder – Input Interface (3 of 4)

getPurchaseOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	dateCancel	Calendar					
Quantity							
	quantityOrdered	BigDecimal					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					

getPurchaseOrder – Input Interface (4 of 4)

getPurchaseOrder: Response Interface

ThegetPurchaseOrder web service operation uses the ShowPurchaseOrder message as the response. These tables list the response interface information for the getPurchaseOrder web service operation:

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
Header		
ShowPurchaseOrder		
Header		
	numberRowsReturned	long
PurchaseOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	businessUnit	String
	deliveryInstruction1	String
	deliveryInstruction2	String
	remark	String
	description1	String
	printMessageCode	String
	adjustmentScheduleCode	String
	itemPriceGroupCode	String
	paymentTermsCode	String
	taXExplanationCode	String
	taxRateAreaCode	String
	taxExemptCertificate	String
	holdCode	String
	isTextAssociated	Boolean
	numberOfInvoices	Integer
	transactionNature	String
	container	String
	freightHandlingCode	String
	zoneNumber	Integer
	isFreightCalculated	Boolean
	modeOfTransportCode	String
	conditionOfTransportCode	String
	activityReasonCode	String
	isFreightApplied	Boolean
	isQuantityPosted	Boolean
	amountGross	Bigdecimal

getPurchaseOrder – Response Interface (1 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
	percentRetainage	String
	ruleRetainage	String
	unitOfMeasureCodeWeight	String
	unitOfMeasureCodeVolume	String
	purgeCode	String
	logicControl	String
	processingModeCode	String
	typeMatch	String
	statusOrderCode	String
	isPrepaid	Boolean
	voucherCode	String
	sendMethodCode	String
	approvedRoutingBy	String
	currencyCodeFrom	String
	currencyModeCode	String
	rateExchangeOverride	BigDecmial
	languageCode	Sttring
	amountOpen	BigDecmial
	currencyCodeBase	String
	timePromisedDelivery	Integer
	timePromisedShip	Integer
	orderedBy	String
	orderTakenBy	String
	entityIdSupplier	Integer
	entityIdShipTo	String
	entityIdBuyer	String
	entityIdCarrier	String
	entityIdMarkFor	String
OriginalOrderKey		
	documentNumber	Integer
	documentCompany	String
	documentTypeCode	String
RelatedOrderKey		
	documentNumber	Integer

getPurchaseOrder – Response Interface (2 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
	documentCompany	String
	documentTypeCode	String
Dates		
	dateRequested	Calendar
	dateTransaction	Calendar
	dateScheduledPick	Calendar
	datePromisedOriginal	Calendar
	datePromisedDelivery	Calendar
	dateShipment	Calendar
	dateCancel	Calendar
	datePriceEffective	Calendar
	datePromisedShip	Calendar
	dateServiceTax	Calendar
	dateAccounting	Calendar
UserReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
Detail		
	PurchaseOrderLineKey	
	documentLineNumber	BigDecimal
	documentSuffix	String
Subledger		
	subledger	String
	subledgerTypeCode	String
	businessUnit	String
	location	String
	lotNumber	String
	description1	String
	description2	String
	lineTypeCode	String
	statusCodeNext	String

getPurchaseOrder – Response Interface (3 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
	statusCodeLast	String
	unitOfMeasureCodeTransaction	String
	overridePriceCode	String
	costMethodPurchasing	String
	printMessageCode	String
	adjustmentScheduleCode	String
	itemPriceGroupCode	String
	priceCategoryLevel	String
	catalogName	String
	discountFactor	String
	paymentTermsCode	String
	taxableCode	String
	container	String
	commodityClassCode	String
	transactionNature	String
	modeOfTransportCode	String
	freightHandlingCode	String
	isFreightCalculated	Boolean
	zoneNumber	Integer
	freightRateCode	String
	freightRateType	String
	conditionOfTransportCode	String
	shippingConditionCode	String
	unitOfMeasureCodePrimary	String
	unitOfMeasureCodeSecondary	String
	unitOfMeasureCodePurchasing	String
	unitWeight	String
	unitVolume	String
	unitOfMeasureCodeVolume	String
	glClassCode	String
	century	String
	yearCode	String
	orderLineStatus	String
	reasonCode	String

getPurchaseOrder – Response Interface (4 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
	isFreightApplied	Boolean
	isQuantityPosted	Boolean
	grossWeight	Bigdecimal
	unitOfMeasureCodeWeight	String
	ledgerTypeCode	String
	serialNumber	Integer
	componentId	Integer
	categoryCodeWorkOrder	String
	ruleRetainage	String
	statusLocationTaxCode	String
	purgeCode	String
	approvedRoutingBy	String
	componentLineNumber	Bigdecimal
	timeChange	Integer
	timeScheduled	Integer
	shiftScheduledCode	String
	currencyCodeFrom	String
	loadNumber	Integer
	reference	String
	promotionId	String
	commodityUnmaskedCode	String
	commodityClassCode	String
	orderLineUniqueKey	String
	entityIdSupplier	Integer
	entityIdShipTo	Integer
	entityIdBuyer	Integer
	entityIdCarrier	Integer
	entityIdMarkFor	Integer
	costUnitPurchasingDomestic	Bigdecimal
	priceExtendedDomestic	Bigdecimal
	amountOnHoldDomestic	Bigdecimal
	amountOpenDomestic	Bigdecimal
	amountReceivedDomestic	Bigdecimal
	amountRelievedDomestic	Bigdecimal

getPurchaseOrder – Response Interface (5 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
	costUnitPurchasingForeign	Bigdecimal
	priceExtendedForeign	Bigdecimal
	amountOnHoldForeign	Bigdecimal
	amountOpenForeign	Bigdecimal
	amountReceivedForeign	Bigdecimal
	amountTaxForeign	Bigdecimal
Item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
OriginalOrderKey		
	documentNumber	Integer
	documentCompany	String
	documentTypeCode	String
OriginalOrderLineKey		
	documentLineNumber	BigDecimal
	documentSuffix	String
RelatedOrderKey		
	documentNumber	Integer
	documentCompany	String
	documentTypeCode	String
RelatedOrderLineKey		
	documentLineNumber	BigDecimal
	documentSuffix	String
ChangeOrderKey		
	documentNumber	Integer
	documentCompany	String
	documentTypeCode	String
ChangeOrderLineKey		
	documentLineNumber	BigDecimal
	documentSuffix	String

getPurchaseOrder – Response Interface (6 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
Dates		
	dateRequested	Calendar
	dateTransaction	Calendar
	dateScheduledPick	Calendar
	datePromisedOriginal	Calendar
	datePromisedDelivery	Calendar
	dateShipment	Calendar
	dateCancel	Calendar
	datePriceEffective	Calendar
	datePromisedShip	Calendar
	dateServiceTax	Calendar
	dateAccounting	Calendar
	dateLotEffective	Calendar
	dateChange	Calendar
Quantity		
	quantityOrdered	Bigdecimal
	quantityChanged	Bigdecimal
	quantityOpen	Bigdecimal
	quantityReceived	Bigdecimal
	quantityCumulative	Bigdecimal
	quantityRelieved	Bigdecimal
	quantityOrderedPrimary	Bigdecimal
	quantityOrderedSecondary	Bigdecimal
GLAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
CategoryCodes		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
	categoryCode006	String

getPurchaseOrder – Response Interface (7 of 8)

ShowPurchaseOrder - Response Interface		
Class	**Field**	**Data Type**
	categoryCode007	String
	categoryCode008	String
	categoryCode009	String
	categoryCode010	String
UserReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String

getPurchaseOrder – Response Interface (8 of 8)

Purchase Order Receipt

This section lists these interface tables:

- processPurchaseOrderReceipt: Input Interface
- processPurchaseOrderReceipt: Response Interface

processPurchaseOrderReceipt: Input Interface

These tables list the input interface information for the processPurchaseOrderReceipt web service operation:

processPurchaseOrderReceipt - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
header		**Header**					
	ProcessingVersion	String					
	supplier	Entity					
	receiptNumber	String	Y				
	rateExchangeOverride	BigDecimal					
purchaseOrderKey		**PurchaseOrderKey**					
	documentNumber	Integer	Y				
	documentTypeCode	String	Y				
	documentCompany	String	Y				
dates		**Dates**					
	dateReceived	Calendar	Y				
	dateAccounting	Calendar	Y				
detail		**Detail**					
purchaseOrderLineKey		**PurchaseOrderLine**					
	documentLineNumber	BigDecimal	Y				
	documentLineNumberSuffix	String					
	receiptLineNumber	BigDecimal	Y				
	remark	String	Y				
	businessunit	String	Y				
	locationReceived	String	Y				
	evaluateReceiptSettlement	String	Y				
item		**ItemSupplierGroup**					
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemSupplier	String					
	itemFreeForm	String					
quantity		**Quantity**					
	quantityReceived	String	Y				
	unitOfMeasureCodeTransaction	String					

processPurchaseOrderReceipt – Input Interface (1 of 2)

processPurchaseOrderReceipt - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
financial		**Financial**					
	costUnitReceivedForeign	BigDecimal	Y				
	costUnitReceivedDomestic	BigDecimal	Y				
	amountReceivedForeign	BigDecimal	Y				
	amountReceivedDomestic	BigDecimal	Y				
delivery		**Delivery**					
	landedCostRuleCode	String	Y				
	shipmentId	String	Y				
	containerId	String	Y				
lot		**Lot**					
	lotNumber	Integer	Y				
	description	String	Y				
	dateLotExpire	Calendar	Y				
	lotAlternative	String	Y				
	lotStatusCode	String	Y				
	lotPotency	Integer	Y				
	lotGrade	String	Y				
serialNumber[]		**SerialNumber**					
	serialNumber	String	Y				
	dateSerialExpire	date	Y				

processPurchaseOrderReceipt – Input Interface (1 of 2)

processPurchaseOrderReceipt: Response Interface

The processPurchaseOrderReceipt web service operation uses the ConfirmPurchaseOrderReceipt message as the response. This table lists the response interface information for the processPurchaseOrderReceipt web service operation:

ConfirmPurchaseOrderReceipt - Response Interface		
Class	Field	Data Type
header		
purchaseOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	receiptNumber	String
	businessUnit	String
	supplier	Entity
detail		
purchaseOrdeLineKey		
	documentLineNumber	BigDecimal
	documentLineNumberSuffix	String
	receiptLineNumber	Integer
	businessUnit	String
E1messages[]		

processPurchaseOrderReceipt – Response Interface

Sales Order

This section lists these interface tables:

- processSalesOrder: Input Interface
- processSalesOrder: Response Interface
- processSalesPriceAdjustment: Input Interface
- processSalesPriceAdjustment: Response Interface
- getItemPriceAndAvailability: Input Interface
- getItemPriceAndAvailability: Response Interface
- getCustomerItemPrice: Input Interface
- getCustomerItemPrice: Response Interface
- getSalesOrder: Input Interface
- getSalesOrder: Response Interface
- getItemListPrice: Input Interface
- getItemListPrice: Response Interface
- getSalesOrderPriceHistory: Input Interface
- getSalesOrderPriceHistory: Response Interface

processSalesOrder: Input Interface

These tables list the input interface information for the processSalesOrder web service operation:

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
header							
processing							
	actionType	String	Y	Y	Y		
	processingVersion	String					
salesOrderKey							
	documentNumber	Integer		Y	Y		Y
	documentTypeCode	String		Y	Y		Y
	documentCompany	String		Y	Y		Y
	businessUnit	String					
	company	String					
	orderedBy	String					
	orderTakenBy	String					
	holdOrderCode	String					
	customerPO	String					
	attachmentText	String					
Entity - soldTo							
	entityId	Integer	Y	Y	Y		
	entityLongId	String					
	entityTaxId	String					
	mailingName	String					
	addressLine1	String					
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
Entity - shipTo							
	entityId	Integer	Y	Y	Y		

processSalesOrder – Input Interface (1 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityLongId	String					
	entityTaxId	String					
	mailingName	String					
	addressLine1	String					
	addressLine2	String					
	addressLine3	String					
	addressLine4	String					
	city	String					
	countyCode	String					
	stateCode	String					
	postalCode	String					
	countryCode	String					
	dateOrdered	Calendar					
	dateRequested	Calendar					
	timeRequested	Integer					
	dateCancel	Calendar					
	dateScheduledPick	Calendar					
	timeScheduledPick	Integer					
	currencyCodeTo	String					
	rateExchange	BigDecimal					
billing							
	adjustmentScheduleCode	String					
	customerPriceGroupCode	String					
	percentDiscountTrade	BigDecimal					
	paymentTermsCode	String					
	paymentInstrumentCode	String					
	printMessageCode	String					
carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

processSalesOrder – Input Interface (2 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	freightHandlingCode	String					
	deliveryInstruction1	String					
	deliveryInstruction2	String					
	taxRateAreaCode	String					
	taxExplanationCode	String					
	creditCardAccountName	String					
	creditCardAccountNumber	String					
	creditCardTypeCode	String					
	checkNumber	String					
	dateExpiration	Calendar					
Entity - deliverTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - invoicedTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - paidBy							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - forwardedTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
userReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					

processSalesOrder – Input Interface (3 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	userReservedReference	String					
detail[]							
processing							
	actionType	String		Y	Y		
	useConfigurationRule	Boolean					
	documentLineNumber	BigDecimal		Y	Y		Y
	businessUnit	String					
	businessUnitDestination	String					
	customerPO	String					
	reference	String					
	agreementId	String					
	agreementSupplement	Integer					
product							
item							
	itemId	Integer	Y				
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	lotNumber	String					
	location	String					
	itemWeight	BigDecimal					
	unitOfMeasureCodeWeight	String					
	itemVolume	BigDecimal					
	unitOfMeasureCodeVolume	String					
	description1	String					
	description2	String					
configuration[]							
itemParent							
	itemId	Integer					
	itemProduct	String					

processSalesOrder – Input Interface (4 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	businessUnit	String					
	quantityComponent	BigDecimal					
	unitOfMeasureCodeComponent	String					
	lineTypeCode	String					
	description1	String					
segment[]							
	segmentNumber	Integer					
	segmentValue	String					
billing							
pricing							
	unitOfMeasureCodePricing	String					
	isZeroPriceOverride	Boolean					
	adjustmentScheduleCode	String					
	datePriceEffective	Calendar					
	priceCode1	String					
	priceCode2	String					
	priceCode3	String					
	priceUnitDomestic	BigDecimal					
	priceUnitForeign	BigDecimal					
	priceExtendedDomestic	BigDecimal					
	priceExtendedForeign	BigDecimal					
	itemPriceGroupCode	String					

processSalesOrder – Input Interface (5 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	paymentTermsCode	String					
	paymentInstrumentCode	String					
	taxRateAreaCode	String					
	taxExplanationCode	String					
	taxableCode	String					
	modeOfTransportCode	String					
	printMessageCode	String					
carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	freightHandlingCode	String					
	subledger	String					
	subledgerTypeCode	String					
	dutyStatusCode	String					
original Order							
originalOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
related Order							
relatedOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
shipTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

processSalesOrder – Input Interface (6 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
supplier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	quantityOrdered	BigDecimal					
	quantityShippable	BigDecimal					
	quantityBackOrdered	BigDecimal					
	quantityCanceled	BigDecimal					
	unitOfMeasureCodeTransaction	String					
	lineTypeCode	String					
	lineOfBusinessCode	String					
	endUseCode	String					
	priorityCode	String					
	dateOrdered	Calendar					
	dateRequested	Calendar					
	timeRequested	Integer					
	dateCancel	Calendar					
	datePromisedShip	Calendar					
	timePromisedShip	Integer					
	dateScheduledPick	Calendar					
	timeScheduledPick	Integer					
	datePromisedOriginal	Calendar					
	timePromisedOriginal	Integer					
	datePromisedDelivery	Calendar					
	timePromisedDelivery	Integer					
	statusCodeLast	String					
	statusCodeNext	String					
categoryCodesSales							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					

processSalesOrder – Input Interface (7 of 8)

processSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	categoryCode004	String					
	categoryCode005	String					
categoryCodesPurchasing							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
userReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					
	userReservedNumber	Integer					
	userReservedReference	String					

processSalesOrder – Input Interface (8 of 8)

processSalesOrder: Response Interface

The processSalesOrder web service operation uses the ConfirmSalesOrder message as the response. These tables list the response interface information for the processSalesOrder web service operation:

ConfirmSalesOrder - Response Interface		
Class	**Field**	**Data Type**
header		
salesOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	actionType	String
	businessUnit	String
	holdOrderCode	String
	amountTotalOrderDomestic	BigDecimal
	amountTotalOrderForeign	BigDecimal
	customerPO	String
	attachmentText	String
dates		
	orderDate	Calendar
financial		
	currencyCode	String
	paymentTerms	String
delivery		
carrier		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	freightHandlingCode	String
Entity - shipTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	mailingName	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String

processSalesOrder – Response Interface (1 of 3)

ConfirmSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	stateCode	String
	postalCode	String
	countryCode	String
detail[]		
	documentLineNumber	BigDecimal
dates		
	datePromisedShip	Calendar
	timePromisedShip	Integer
	datePromisedDelivery	Calendar
	timePromisedDelivery	Integer
	dateRequested	Calendar
	timeRequested	Integer
product		
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
	lotNumber	String
	statusCodeLast	String
	statusCodeNext	String
	lineTypeCode	String
quantity		
	quantityOrdered	BigDecimal
	quantityShippable	BigDecimal
	quantityBackOrdered	BigDecimal
	quantityCanceled	BigDecimal
	unitOfMeasureCodeTransaction	String
financial		
	priceUnitDomestic	BigDecimal
	priceUnitForeign	BigDecimal
	priceExtendedDomestic	BigDecimal
	priceExtendedForeign	BigDecimal

processSalesOrder – Response Interface (2 of 3)

ConfirmSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	unitOfMeasureCodePricing	String
Entity - shipTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
delivery		
	modeOfTransportCode	String
carrier		Entity
	entityId	Integer
	entityLongId	String
	entityTaxId	String
messages[]		
	messages	String

processSalesOrder – Response Interface (3 of 3)

processSalesPriceAdjustment: Input Interface

This table lists the input interface information for the processSalesPriceAdjustment web service operation:

processSalesPriceAdjustment - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
customer		Entity					
	entityId	Integer		Y			Y
	entityLongId	String					
	entityTaxId	String					
item		ItemGroupCusto					
	itemId	Integer		Y			Y
	itemProduct	String		Y			
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
financial		Financial					
	currencyCode	String		Y			Y
	amountAdjustment	BigDecimal		Y			
processing		Processing					
	actionType	String	Y	Y			
	processingVersion	String		Y			
dates		Dates					
	dateExpiration	Date		Y			
	dateEffective	Date		Y			
	costTypeCode	String					
	adjustmentTypeCode	String		Y			Y
	unitOfMeasureCodeTransaction	String	Y	Y			Y
	promotionAddChangeMask	String					
	priceAdjustmentId	Integer		Y			Y

processSalesPriceAdjustment – Input Interface

processSalesPriceAdjustment: Response Interface

The processSalesPriceAdjustment web service operation uses the ConfirmSalesPriceAdjustment message as the response. This table lists the response interface information for the processSalesPriceAdjustment web service operation:

ConfirmSalesPriceAdjustment - Response Interface		
Class	**Field**	**Data Type**
customer		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	itemFreeForm	String
financial		
	currencyCode	String
	amountAdjustment	BigDecimal
processing		
	actionType	String
	processingVersion	String
	unitOfMeasureCodeTransaction	String
	adjustmentTypeCode	String
	costTypeCode	String
	priceAdjustmentId	Integer
messages[]		
	messages	String

processSalesPriceAdjustment – Response Interface

getItemPriceAndAvailability: Input Interface

This table lists the input interface information for the getItemPriceAndAvailability web service operation:

getItemPriceAndAvailability - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
processing							
	processingVersion	String				Y	
	businessUnit	String				Y	
customer							
	entityId	Integer				Y	
	entityLongId	String				Y	
	entityTaxId	String				Y	
	currencyCode	String				Y	
	adjustmentScheduleCode	String				Y	
product							
item							
	itemId	Integer				Y	
	itemProduct	String				Y	
	itemCatalog	String				Y	
	itemFreeForm	String				Y	
	itemCustomer	String				Y	
	quantityOrdered	BigDecimal				Y	
	unitOfMeasureCodeTransaction	String				Y	

getItemPriceAndAvailabilit – Input Interface

getItemPriceAndAvailability: Response Interface

The getItemPriceAndAvailability web service operation uses the ShowItemPriceAndAvailability message as the response. This table lists the response interface information for the getItemPriceAndAvailability web service operation:

ShowItemPriceAndAvailability - Response Interface		
Class	**Field**	**Data Type**
	currencyCode	String
product		
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemCustomer	String
	quantityOrdered	BigDecimal
	unitOfMeasureCodeTransaction	String
	priceUnit	BigDecimal
	priceExtended	BigDecimal
	unitOfMeasureCodePricing	String
availability[]		
	quantityAvailable	BigDecimal
warehouse		
	warehouse	Integer
	warehouseDescriptive	String
address		
	mailingName	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
messages[]		
	messages	String

getItemPriceAndAvailabilit – Response Interface

getCustomerItemPrice: Input Interface

These tables list the input interface information for the getCustomerItemPrice web service operation:

getCustomerItemPrice - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
processing							
	processingVersion	String				Y	
customer							
soldTo							
	entityId	Integer				Y	Y
	entityLongId	String				Y	
	entityTaxId	String				Y	
shipTo							
	entityId	Integer				Y	
	entityLongId	String				Y	
	entityTaxId	String				Y	
	businessUnit	String				Y	
	priceAdjustmentId	String				Y	
	currencyCode	String				Y	
	rateExchange	BigDecimal				Y	
carrier		**Entity**					
	entityId	Integer				Y	
	entityLongId	String				Y	
	entityTaxId	String				Y	
	freightHandlingCode	String				Y	
	discountTrade	BigDecimal				Y	
	customerPriceGroupCode	String				Y	
	zoneNumber	String				Y	
	routeCode	String				Y	
	stopCode	String				Y	
product							
item							
	itemId	Integer				Y	Y
	itemProduct	String				Y	Y
	itemCatalog	String				Y	Y
	itemFreeForm	String				Y	Y

getCustomerItemPrice – Input Interface (1 of 2)

getCustomerItemPrice - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	itemCustomer	String				Y	Y
	businessUnit	String				Y	Y
ship To							
	entityId	Integer				Y	
	entityLongId	String				Y	
	entityTaxId	String				Y	
	lineType	String				Y	
	lotNumber	String				Y	
	location	String				Y	
	unitOfMeasureCodeVolume	String				Y	
	unitOfMeasureCodeWeight	String				Y	
	itemVolume	BigDecimal				Y	
	itemWeight	BigDecimal				Y	
	transactionQuantity	BigDecimal				Y	Y
	unitOfMeasureCodeTransaction	String				Y	
	unitOfMeasureCodePricing	String				Y	
	datePriceEffective	Date				Y	
	agreementID	String				Y	
	paymentTermsCode	String				Y	
	paymentInstrumentCode	String				Y	
	modeOfTransportCode	String				Y	
	statusCodeDuty	String				Y	
	endUseCode	String				Y	
	lineOfBusinessCode	String				Y	
	priceCode1	String				Y	
	priceCode2	String				Y	
	priceCode3	String				Y	

getCustomerItemPrice – Input Interface (2 of 2)

getCustomerItemPrice: Response Interface

The getCustomerItemPrice web service operation uses the ShowCustomerItemPrice message as the response. These tables list the response interface information for the getCustomerItemPrice web service operation:

ShowCustomerItemPrice - Response Interface		
Class	**Field**	**Data Type**
customer		
soldTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
shipTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	businessUnit	String
	currencyCode	String
	currencyModeCode	String
product		
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemFreeForm	String
	itemCustomer	String
	businessUnit	String
ship To		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	lineTypeCode	String
	lotNumber	String
	location	String
	unitOfMeasureCodeVolume	String
	unitOfMeasureCodeWeight	String
	itemVolume	BigDecimal
	itemWeight	BigDecimal
	quantityOrdered	BigDecimal
	unitOfMeasureCodeTransaction	String
	unitOfMeasureCodePricing	String

getCustomerItemPrice – Response Interface (1 of 2)

ShowCustomerItemPrice - Response Interface		
Class	**Field**	**Data Type**
	datePriceEffective	Date
	priceUnitDomestic	BigDecimal
	priceUnitForeign	BigDecimal
	priceExtendedDomestic	BigDecimal
	priceExtendedForeign	BigDecimal
	priceListDomestic	BigDecimal
	priceListForeign	BigDecimal
	costUnitDomestic	BigDecimal
	costUnitForeign	BigDecimal
	costExtendedDomestic	BigDecimal
	costExtendedForeign	BigDecimal
messages[]		
	messages	String

getCustomerItemPrice – Response Interface (2 of 2)

getSalesOrder: Input Interface

These tables list the input interface information for the getSalesOrder web service operation:

getSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
header							
salesOrderKey							
	documentNumber	Integer					Y
	documentTypeCode	String					Y
	documentCompany	String					Y
	businessUnit	String					
	company	String					
	orderTakenBy	String					
	holdOrderCode	String					
	customerPO	String					
	amountTotalOrderDomestic	BigDecimal					
	amountTotalOrderForeign	BigDecimal					
soldTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
shipTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	dateTransaction	Calendar					
	dateRequested	Calendar					
	dateCancel	Calendar					
	datePromisedPick	Calendar					
	dateActualShip	Calendar					
	currencyCodeFrom	String					
	rateExchangeOverride	BigDecimal					
deliverTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					

getSalesOrder – Input Interface (1 of 5)

getSalesOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
invoicedTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
paidBy							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
forwardedTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
originalOrder							
originalOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	RMAId	Integer					
	RMAType	String					
detail							
	documentLineNumber	BigDecimal					Y
	businessUnit	String					
	company	String					
	customerPO	String					
soldTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
shipTo							
	entityId	Integer					
	entityLongId	String					

getSalesOrder – Input Interface (2 of 5)

getSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityTaxId	String					
	description1	String					
	description2	String					
	agreementId	String					
parent							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
	shipmentId	Integer					
	pickSlipId	Integer					
	deliveryId	Integer					
product							
item							
	itemId	Integer					
	itemProduct	String					
	itemCatalog	String					
	itemCustomer	String					
	itemFreeForm	String					
	lineTypeCode	String					
	statusCodeLast	String					
	statusCodeNext	String					
	lotNumber	String					
	location	String					
	zoneNumber	String					
carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
originalOrder							
originalOrderKey							
	documentNumber	String					
	documentTypeCode	String					

getSalesOrder – Input Interface (3 of 5)

getSalesOrder - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	documentCompany	String					
relatedOrder							
relatedOrderKey							
	documentNumber	String					
	documentTypeCode	String					
	documentCompany	String					
	documentLineNumber	BigDecimal					
	dateTransaction	Calendar					
	dateRequested	Calendar					
	dateActualShip	Calendar					
	dateScheduledPick	Calendar					
	datePromisedOriginal	Calendar					
	dateInvoice	Calendar					
	dateCancel	Calendar					
	dateAccounting	Calendar					
	datePromisedDelivery	Calendar					
	datePriceEffective	Calendar					
	freightHandlingCode	String					
	modeOfTransportCode	String					
	unitOfMeasureCodeTransaction	String					
	unitOfMeasureCodePricing	String					
	priceUnitDomestic	BigDecimal					
	priceUnitForeign	BigDecimal					
	currencyCodeFrom	String					
	rateExchangeOverride	BigDecimal					
	quantityOrdered	BigDecimal					
	quantityShippable	BigDecimal					
	quantityBackOrdered	BigDecimal					
	quantityCanceled	BigDecimal					
deliverTo							
	entityId	Integer					
	entityLongId	String					

getSalesOrder – Input Interface (4 of 5)

getSalesOrder - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityTaxId	String					
document							
documentKey							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
categoryCodesSales							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					
categoryCodesPurchasing							
	categoryCode001	String					
	categoryCode002	String					
	categoryCode003	String					
	categoryCode004	String					
	categoryCode005	String					

getSalesOrder – Input Interface (5 of 5)

getSalesOrder: Response Interface

The getSalesOrder web service operation uses the ShowSalesOrder message as the response. These tables list the response interface information for the getSalesOrder web service operation:

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
header[]		
salesOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	businessUnit	String
	company	String
	orderedBy	String
	orderTakenBy	String
	holdOrderCode	String
	customerPO	String
	printMessageCode	String
	amountTotalOrderDomestic	BigDecimal
	amountTotalOrderForeign	BigDecimal
sold To		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
shipTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	mailingName	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	stateCode	String
	city	String
	countyCode	String
	postalCode	String
	countryCode	String

getSalesOrder – Response Interface (1 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	dateTransaction	Calendar
	dateRequested	Calendar
	dateCancel	Calendar
	datePromisedShip	Calendar
	dateScheduledPick	Calendar
	dateActualShip	Calendar
	currencyCodeFrom	String
	currencyModeCode	String
	rateExchangeOverride	BigDecimal
	paymentTermsCode	String
	entityIdCarrier	Integer
	adjusmentScheduleCode	String
	freightHandlingCode	String
	paymentInsttrumentCode	String
	taxRateAreaCode	String
	taxExplanationCode	String
	percentDiscountTrade	BigDecimal
	customerPriceGroupCode	String
	deliveryInstruction1	String
	deliveryInstruction2	String
	routCode	String
	StopCode	String
	zoneNumber	String
	modeOfTransportCode	String
	reasonCode	String
deliverTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
invoiceTo		
	entityId	Integer
	entityLongId	String

getSalesOrder – Response Interface (2 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	entityTaxId	String
paidBy		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
forwardedTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
originalOrder		
originalOrderKey		
	documentNumber	String
	documentTypeCode	String
	documentCompany	String
	RMAId	Integer
	RMAType	String
userReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
detail[]		
	documentLineNumber	BigDecimal
	businessUnit	String
	company	String
	customerPO	String
soldTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String

getSalesOrder – Response Interface (3 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
shipTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	reference	String
	agreementId	String
	agreementSupplement	Integer
parent		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	shipmentId	Integer
	pickSlipId	Integer
	deliveryId	Integer
	currencyCodeFrom	String
	rateExchangeOverride	BigDecimal
product		
item		
	itemId	Integer
	itemProduct	String
	itemCatalog	String
	itemUPCEAN	String
	lotNumber	String
	location	String
	zoneNumber	String
	itemWeight	BigDecimal
	unitOfMeasureCodeWeight	String
	itemVolume	BigDecimal
	unitOfMeasureCodeVolume	String
	description1	String
	description2	String

getSalesOrder – Response Interface (4 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
lineBilling		
linePricing		
	unitOfMeasureCodePricing	String
	adjusmentScheduleCode	String
	datePriceEffective	Calendar
	priceCode1	String
	priceCode2	String
	priceCode3	String
	priceUnitDomestic	BigDecimal
	priceUnitForeign	BigDecimal
	priceExtendedDomestic	BigDecimal
	priceExtendedForeign	BigDecimal
	itemPriceGroupCode	String
	paymentTermsCode	String
	paymentInstrumentCode	String
	taxRateAreaCode	String
	taxExplanationCode	String
	taxableCode	String
	modeOfTransportCode	String
	printMessageCode	String
carrier		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
	freightHandlingCode	String
	subledger	String
	subledgerTypeCode	String
	dutyStatusCode	String
originalOrder		
originalOrderKey		
	documentNumber	String
	documentTypeCode	String

getSalesOrder – Response Interface (5 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	documentCompany	String
	documentLineNumber	BigDecimal
relatedOrder		
relatedOrderKey		
	documentNumber	String
	documentTypeCode	String
	documentCompany	String
	documentLineNumber	BigDecimal
	entityIdSupplier	Integer
	quantityOrdered	BigDecimal
	quantityShippable	BigDecimal
	quantityBackOrdered	BigDecimal
	quantityCanceled	BigDecimal
	unitOfMeasureCodeTransaction	String
	costUnitPurchasingDomestic	BigDecimal
	costUnitPurchasingForeign	BigDecimal
	costExtendedDomestic	BigDecimal
	costExtendedForeign	BigDecimal
	lineTypeCode	String
	lineOfBusinessCode	String
	endUseCode	String
	dateOrdered	Calendar
	dateRequested	Calendar
	timeRequested	Integer
	dateCancel	Calendar
	datePromisedShip	Calendar
	timePromisedShip	Integer
	dateScheduledPick	Calendar
	timeScheduledPick	Integer
	datePromisedOriginal	Calendar
	timePromisedOriginal	Integer
	datePromisedDelivery	Calendar

getSalesOrder – Response Interface (6 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	timePromisedDelivery	Integer
	dateActualShip	Calendar
	dateInvoice	Calendar
	dateAccounting	Calendar
	statusCodeLast	String
	statusCodeNext	String
deliverTo		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
document		
documentKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
userReservedData		
	userReservedCode	String
	userReservedDate	Calendar
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
categoryCodesSales		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String
	categoryCode005	String
categoryCodesPurchasing		
	categoryCode001	String
	categoryCode002	String
	categoryCode003	String
	categoryCode004	String

getSalesOrder – Response Interface (7 of 8)

ShowSalesOrder - Response Interface		
Class	**Field**	**Data Type**
	categoryCode005	String
messages[]		
	messages	String

getSalesOrder – Response Interface (8 of 8)

getItemListPrice: Input Interface

This table lists the input interface information for the getItemListPrice web service operation:

getItemListPrice - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	businessUnit	String				Y	
customer							
	entityId	Integer				Y	
	entityLongId	String				Y	
	entityTaxId	String				Y	
item							
	itemId	Integer				Y	Y
	itemProduct	String				Y	
	itemCatalog	String				Y	
	itemFreeForm	String				Y	
	itemCustomer	String				Y	
	unitOfMeasureCode	String				Y	
	dateEffective	Calendar				Y	
	dateExpiration	Calendar				Y	
	currencyCode	String				Y	

getItemListPrice – Input Interface

getItemListPrice: Response Interface

The getItemListPrice web service operation uses the ShowItemListPrice message as the response. This table lists the response interface information for the getItemListPrice web service operation:

ShowItemListPrice - Response Interface		
Class	Field	Data Type
showItemListPrice[]		
	businessUnit	String
	entityIdCustomer	Integer
	itemId	Integer
	unitOfMeasureCode	String
	lotNumber	String
	location	String
	lotGrade	String
	lotPotency	String
	dateEffective	Calendar
	dateExpiration	Calendar
	currencyCode	String
	priceList	BigDecimal
messages[]		
	messages	String

getItemListPrice – Response Interface

getSalesOrderPriceHistory: Input Interface

This table lists the input interface information for the getSalesOrderPriceHistory web service operation:

getSalesOrderPriceHistory - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
detail							
salesOrderKey							
	documentNumber	Integer				Y	Y
	documentTypeCode	String				Y	Y
	documentCompany	String				Y	Y
	documentLineNumber	BigDecimal				Y	Y

getSalesOrderPriceHistory – Input Interface

getSalesOrderPriceHistory: Response Interface

The getSalesOrderPriceHistory web service operation uses the ShowSalesOrderPriceHistory message as the response. This table lists the response interface information for the getSalesOrderPriceHistory web service operation:

ShowSalesOrderPriceHistory - Response Interface		
Class	Field	Data Type
	numberOfRowsReturned	Long
detail[]		
salesOrderKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
	documentLineNumber	BigDecimal
priceAdjustmentLedger[]		
	promotionDisplayControl	String
	adjustmentGroup	String
	adjustmentMutuallyExclusive	String
	adjustmentCalculation	String
	valueBasedOn	BigDecimal
	manualDiscount	String
	overridePriceCode	String
	ledgerTypeCode	String
	adjustmentReasonCode	String
	unitOfMeasureCodeFactorValue	String
	basisCode	String
	amountAdjustmentForeign	BigDecimal
	amountAdjustmentDomestic	BigDecimal
	valueFactor	BigDecimal
	adjustmentTypeCode	String
	sequenceNumber	Integer
	quantityFrom	BigDecimal
	adjustmentScheduleCode	String
messages[]		
	messages	String

getSalesOrderPriceHistory – Response Interface

Supplier

This section lists these interface tables:

- processSupplier: Input Interface
- processSupplier: Response Interface
- getSupplier: Input Interface
- getSupplier: Response Interface

processSupplier: Input Interface

These tables list the input interface information for the processSupplier web service operation:

processSupplier - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
Supplier							
	freightHandlingCode	String					
	glOffsetCode	String					
	addressTypeCodeBilling	String					
	creditMessageCode	String					
	holdOrderCode	String					
	holdPaymentCode	String					
	minimumCheckAmountCode	String					
	customerPriceGroupCode	String					
	categoryCodePurchasing	String					
	adjustmentScheduleCode	String					
	orderValueMaximum	Integer					
	orderValueMinimum	Integer					
	deliveryInstruction1	String					
	deliveryInstruction2	String					
	taxPercentWithholding	Integer					
	taxRateAreaCodeWithholding	String					
	taxExplanationCodeWithholding	String					
	currencyCodeAddressBook	String					
	floatDaysForPayments	Integer					
	orderTemplateCode	String					
	printMessageCode	String					
	numberOfInvoices	Integer					
	showPricePickList	Boolean					
	evaluateReceiptSettlement	String					
	specialInstruction1	String					
	specialInstruction2	String					
	specialInstruction3	String					
	specialInstruction4	String					
	specialInstruction5	String					
	itemRestrictionsCode	String					

processSupplier – Input Interface (1 of 4)

processSupplier - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
	unitOfMeasureCodeWeight	String					
	unitOfMeasureCodeVolume	String					
	preNoteCode	String					
	sequenceLedgerCode	String					
	amountVoucheredPYE	BigDecimal					
	amountVoucheredYTD	BigDecimal					
	amountAddressBook	BigDecimal					
	amountOpenOrders	BigDecimal					
	revenueNetted	Boolean					
	ownerCode	Boolean					
	daysTransit	Integer					
	zoneNumber	String					
	stopCode	String					
	routeCode	String					
	programId	String					
Processing							
	actionType	String	Y	Y	Y		
	processingVersion	String					
Entity - Approver							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Financial							
	currencyCodeAccountsPayable	String					
	paymentInstrumentCode	String					
	paymentTermsCode	String					
Tax							
	taxRateAreaCode	String					
	taxExplanationCode	String					
Entity - TaxAuthorityWithholding							
	entityId	Integer					

processSupplier – Input Interface (2 of 4)

processSupplier - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	entityLongId	String					
	entityTaxId	String					
Entity - Carrier							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - CarrierParent							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
Entity - RelatedEntity							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
DefaultDistributionAccount							
GLAccountKey							
	accountId	Integer					
	accountLongId	String					
	accountAlternate	String					
GLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					
ModelJournalEntry							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					
UserReservedData							
	userReservedCode	String					
	userReservedDate	Calendar					
	userReservedAmount	BigDecimal					

processSupplier – Input Interface (3 of 4)

processSupplier - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	userReservedNumber	Integer					
	userReservedReference	String					
ProcessAddressBook - Address							
PhoneNumber[] -							
ElectronicAddress[]							

processSupplier – Input Interface (4 of 4)

processSupplier: Response Interface

The processSupplier web service operation uses the ConfirmProcessSupplier message as the response. This table lists the response interface information for the processSupplier web service operation:

ConfirmProcessSupplier - Response Interface		
Class	**Field**	**Data Type**
AddressBookOutput		
	entityName	String
	entityTypeCode	String
	businessUnit	String
	industryClassificationCode	String
	languageCode	String
entity		
	EntityId	Integer
	EntityLongId	String
	EntityTaxId	String
PhoneNumber[]		
	actionType	String
	contactId	Integer
	phoneLineNumber	Integer
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddress[]		
	actionType	String
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	String
messages[]		
	messages	String

processSupplier – Response Interface

getSupplier: Input Interface

These tables lists the input interface information for the getSupplier web service operation:

getSupplier - Input Interface			Required				Key
Class	Field	Data Type*	A	C	D/C	I	(Y/N)
	entityTypeCode	String					
	entityName	String					
	businessUnit	String					
	industryClassificationCode	String					
	creditMessageCode	String					
	holdOrderCode	String					
	holdPaymentCode	String					
	glOffsetCode	String					
	languageCode	String					
entity							
	entityId	Integer					Y
	entityLongId	String					
	entityTaxId	String					
addressInput							
	countyCode	String					
	stateCode	String					
	countryCode	String					
	postalCode	String					
financial							
	currencyCodeAccountsPayable	String					
	paymentInstrumentCode	String					
	paymentTermsCode	String					
defaultDistributionAccount							
gLAccountKey							
	accountId	String					
	accountLongId	String					
	accountAlternate	String					
gLAccount							
	businessUnit	String					
	objectAccount	String					
	subsidiary	String					

getSupplier – Input Interface (1 of 2)

getSupplier - Input Interface			Required				Key
Class	Field	Data Type*	A	C	D/C	I	(Y/N)
modelJournalEntry							
	documentNumber	Integer					
	documentTypeCode	String					
	documentCompany	String					

getSupplier – Input Interface (2 of 2)

getSupplier: Response Interface

The getSupplier web service operation uses the ShowSupplier message as the response. This table lists the response interface information for the getSupplier web service operation:

ShowSupplier - Response Interface		
Class	**Field**	**Data Type**
ShowSupplierData[]		
	entityIdApprover	Integer
userReservedData		
	userReservedCode	String
	userReservedDate	Date
	userReservedAmount	BigDecimal
	userReservedNumber	Integer
	userReservedReference	String
defaultDistributionAccount		
gLAccountKey		
	accountId	Integer
	accountLongId	String
	accountAlternate	String
gLAccount		
	businessUnit	String
	objectAccount	String
	subsidiary	String
modelJournalEntry		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
financial		
	currencyCodeAccountsPayable	String
	paymentInstrumentCode	String
	paymentTermsCode	String
tax		
	taxRateAreaCode	String
	taxExplanationCode	String
	entityIdCarrier	Integer
	entityIdCarrierParent	Integer
	freightHandlingCode	String
	glOffsetCode	String
	entityIdRelated	Integer
	addressTypeCodeBilling	String

getSupplier – Response Interface (1 of 4)

ShowSupplier - Response Interface		
Class	**Field**	**Data Type**
	creditMessageCode	String
	holdOrderCode	String
	holdPaymentCode	String
	minimumCheckAmountCode	String
	customerPriceGroupCode	String
	categoryCodePurchasing	String
	adjustmentScheduleCode	String
	orderValueMaximum	Integer
	orderValueMinimum	Integer
	deliveryInstruction1	String
	deliveryInstruction2	String
	entityIdTaxAuthorityWithholding	Integer
	taxPercentWithholding	Integer
	taxRateAreaCodeWithholding	String
	taxExplanationCodeWithholding	String
	currencyCodeAddressBook	String
	floatDaysForPayments	Integer
	orderTemplateCode	String
	printMessageCode	String
	numberOfInvoices	Integer
	showPricePickList	String
	evaluateReceiptSettlement	String
	specialInstruction1	String
	specialInstruction2	String
	specialInstruction3	String
	specialInstruction4	String
	specialInstruction5	String
	itemRestrictionsCode	String
	unitOfMeasureCodeWeight	String
	unitOfMeasureCodeVolume	String
	preNoteCode	String
	sequenceLedgerCode	String
	multiplePaymentsCode	String
	amountVoucheredPYE	BigDecimal

getSupplier – Response Interface (2 of 4)

ShowSupplier - Response Interface		
Class	**Field**	**Data Type**
	amountVoucheredYTD	BigDecimal
	amountAddressBook	BigDecimal
	amountOpenOrders	Integer
	revenueNetted	String
	deliveryNote	String
	routeCode	String
	stopCode	String
	zoneNumber	String
	daysTransit	String
	ownerCode	String
showSupplierAddressBook		
entity		
	entityId	Integer
	entityLongId	String
	entityTaxId	String
addressSupplier		
	mailingName	String
	mailingNameSecondary	String
	addressLine1	String
	addressLine2	String
	addressLine3	String
	addressLine4	String
	city	String
	countyCode	String
	stateCode	String
	postalCode	String
	countryCode	String
	dateEffective	Date
	businessUnit	String
	entityIdFactorSpecialPayee	Integer
	entityIdParent	String
	remark	String
	personCorporationCode	String
	industryClassificationCode	String

getSupplier – Response Interface (3 of 4)

ShowSupplier - Response Interface		
Class	**Field**	**Data Type**
	languageCode	String
	entityTaxIdAdditional	String
	entityTypeCode	String
	entityName	String
PhoneNumber[]		
	contactId	Integer
	relatedPersonId	String
	phoneLineNumber	String
	areaCode	String
	phoneTypeCode	String
	phoneNumber	String
ElectronicAddress[]		
	contactId	Integer
	electronicAddressLineNumber	Integer
	electronicAddressTypeCode	String
	electronicAddress	String
	electronicAddressClassificationCode	String
	messageIndicatorCode	Integer
Messages[]		

getSupplier – Response Interface (4 of 4)

Voucher Match

This section lists these interface tables:

- processVoucherMatch: Input Interface
- processVoucherMatch: Response Interface

processVoucherMatch: Input Interface

These tables list the input interface information for the processVoucherMatch web service operation:

processVoucherMatch - Input Interface			Required				Key
Class	**Field**	**Data Type**	**A**	**C**	**D/C**	**I**	**(Y/N)**
header		**Header**	Y	Y	Y	Y	
	processingVersion	String					
	supplierInvoiceNumber	String	Y				
	company	String	Y				
supplier							
	entityId	Integer	Y				
	entityLongId	String					
	entityTaxId	String					
remitTo							
	entityId	Integer					
	entityLongId	String					
	entityTaxId	String					
purchaseOrderKey							
	documentNumber	Integer	Y				
	documentTypeCode	String	Y				
	documentCompany	String	Y				
dates							
	dateInvoice	Calendar					
	dateAccounting	Calendar					
financialHeader							
	currencyCode	String	Y				
	rateExchangeOverride	BigDecimal					
	amountTotalInvoice	BigDecimal	Y				
	amountTax	BigDecimal					
	paymentTermsCode	String					
additionalCharge[]							
	additionalChargeAmt	BigDecimal					
	additionalChargeQty	BigDecimal					
	lineTaxableCode	String					
	expenseAcctNumber	String					

processVoucherMatch – Input Interface (1 of 2)

processVoucherMatch - Input Interface			Required				Key
Class	Field	Data Type	A	C	D/C	I	(Y/N)
	taxExplanationCode	String					
	taxRateArea	String					
	lineTaxAmount	BigDecimal					
	description1	String					
detail[]							
	checkRemark	String					
	datePaymentDue	Calendar					
purchaseOrderLineKey							
	documentLineNumber	BigDecimal	Y				
	documentLineNumberSuffix	String	Y				
item							
	itemId	Integer					
	itemProduct	String	Y				
	itemCatalog	String					
	itemSupplier	String					
	itemFreeForm	String					
quantity							
	unitOfMeasureCodeTransaction	String	Y				
	quantityToPay	BigDecimal	Y				
financialDetail							
	costUnit	BigDecimal	Y				
	costExtended	BigDecimal	Y				
	amountTax	BigDecimal					

processVoucherMatch – Input Interface (2 of 2)

processVoucherMatch: Response Interface

The processVoucherMatch web service operation uses the Confirm VoucherMatch message as the response. This table lists the response interface information for the processVoucherMatch web service operation:

ConfirmVoucherMatch - Response Interface		
Class	Field	Data Type
header		
	SupplierInvoiceNumber	String
voucherKey		
	documentNumber	Integer
	documentTypeCode	String
	documentCompany	String
details[]		
voucherLineKey		
	documentLineNumber	BigDecimal
	documentLineNumberSuffix	String
E1messages[]		

processVoucherMatch – Response Interface

Glossary of JD Edwards EnterpriseOne Terms

Accessor Methods/Assessors Java methods to "get" and "set" the elements of a value object or other source file.

activity rule The criteria by which an object progresses from one given point to the next in a flow.

add mode A condition of a form that enables users to input data.

Advanced Planning Agent (APAg) A JD Edwards EnterpriseOne tool that can be used to extract, transform, and load enterprise data. APAg supports access to data sources in the form of rational databases, flat file format, and other data or message encoding, such as XML.

alternate currency A currency that is different from the domestic currency (when dealing with a domestic-only transaction) or the domestic and foreign currency of a transaction.

In JD Edwards EnterpriseOne Financial Management, alternate currency processing enables you to enter receipts and payments in a currency other than the one in which they were issued.

Application Server Software that provides the business logic for an application program in a distributed environment. The servers can be Oracle Application Server (OAS) or WebSphere Application Server (WAS).

as if processing A process that enables you to view currency amounts as if they were entered in a currency different from the domestic and foreign currency of the transaction.

as of processing A process that is run as of a specific point in time to summarize transactions up to that date. For example, you can run various JD Edwards EnterpriseOne reports as of a specific date to determine balances and amounts of accounts, units, and so on as of that date.

Auto Commit Transaction A database connection through which all database operations are immediately written to the database.

back-to-back process A process in JD Edwards EnterpriseOne Supply Management that contains the same keys that are used in another process.

batch processing A process of transferring records from a third-party system to JD Edwards EnterpriseOne.

In JD Edwards EnterpriseOne Financial Management, batch processing enables you to transfer invoices and vouchers that are entered in a system other than JD Edwards EnterpriseOne to JD Edwards EnterpriseOne Accounts Receivable and JD Edwards EnterpriseOne Accounts Payable, respectively. In addition, you can transfer address book information, including customer and supplier records, to JD Edwards EnterpriseOne.

batch server A server that is designated for running batch processing requests. A batch server typically does not contain a database nor does it run interactive applications.

batch-of-one immediate A transaction method that enables a client application to perform work on a client workstation, then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks.

See also direct connect and store-and-forward.

best practices Non-mandatory guidelines that help the developer make better design decisions.

BPEL Abbreviation for *Business Process Execution Language*, a standard web services orchestration language, which enables you to assemble discrete services into an end-to-end process flow.

BPEL PM Abbreviation for *Business Process Execution Language Process Manager*, a comprehensive infrastructure for creating, deploying, and managing BPEL business processes.

Build Configuration File Configurable settings in a text file that are used by a build program to generate ANT scripts. ANT is a software tool used for automating build processes. These scripts build published business services.

build engineer An actor that is responsible for building, mastering, and packaging artifacts. Some build engineers are responsible for building application artifacts, and some are responsible for building foundation artifacts.

Build Program A WIN32 executable that reads build configuration files and generates an ANT script for building published business services.

business analyst An actor that determines if and why an EnterpriseOne business service needs to be developed.

business function A named set of user-created, reusable business rules and logs that can be called through event rules. Business functions can run a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the application programming interfaces (APIs) that enable them to be called from a form, a database trigger, or a non-JD Edwards EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules, and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.

business function event rule See named event rule (NER).

business service EnterpriseOne business logic written in Java. A business service is a collection of one or more artifacts. Unless specified otherwise, a business service implies both a published business service and business service.

business service artifacts Source files, descriptors, and so on that are managed for business service development and are needed for the business service build process.

business service class method A method that accesses resources provided by the business service framework.

business service configuration files Configuration files include, but are not limited to, interop.ini, JDBj.ini, and jdelog.properties.

business service cross reference A key and value data pair used during orchestration. Collectively refers to both the code and the key cross reference in the WSG/XPI based system.

business service cross-reference utilities Utility services installed in a BPEL/ESB environment that are used to access JD Edwards EnterpriseOne orchestration cross-reference data.

business service development environment A framework needed by an integration developer to develop and manage business services.

business services development tool Otherwise known as JDeveloper.

business service EnterpriseOne object A collection of artifacts managed by EnterpriseOne LCM tools. Named and represented within EnterpriseOne LCM similarly to other EnterpriseOne objects like tables, views, forms, and so on.

business service framework Parts of the business service foundation that are specifically for supporting business service development.

business service payload An object that is passed between an enterprise server and a business services server. The business service payload contains the input to the business service when passed to the business services server. The business service payload contains the results from the business service when passed to the Enterprise Server. In the case of notifications, the return business service payload contains the acknowledgement.

business service property Key value data pairs used to control the behavior or functionality of business services.

Business Service Property Admin Tool An EnterpriseOne application for developers and administrators to manage business service property records.

business service property business service group A classification for business service property at the business service level. This is generally a business service name. A business service level contains one or more business service property groups. Each business service property group may contain zero or more business service property records.

business service property categorization A way to categorize business service properties. These properties are categorized by business service.

business service property key A unique name that identifies the business service property globally in the system.

business service property utilities A utility API used in business service development to access EnterpriseOne business service property data.

business service property value A value for a business service property.

business service repository A source management system, for example ClearCase, where business service artifacts and build files are stored. Or, a physical directory in network.

business services server The physical machine where the business services are located. Business services are run on an application server instance.

business services source file or business service class One type of business service artifact. A text file with the .java file type written to be compiled by a Java compiler.

business service value object template The structural representation of a business service value object used in a C-business function.

Business Service Value Object Template Utility A utility used to create a business service value object template from a business service value object.

business services server artifact The object to be deployed to the business services server.

business view A means for selecting specific columns from one or more JD Edwards EnterpriseOne application tables whose data is used in an application or report. A business view does not select specific rows, nor does it contain any actual data. It is strictly a view through which you can manipulate data.

central objects merge A process that blends a customer's modifications to the objects in a current release with objects in a new release.

central server A server that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. In a typical JD Edwards EnterpriseOne installation, the software is loaded on to one machine—the central server. Then, copies of the software are pushed out or downloaded to various workstations attached to it. That way, if the software is altered or corrupted through its use on workstations, an original set of objects (central objects) is always available on the central server.

charts Tables of information in JD Edwards EnterpriseOne that appear on forms in the software.

check-in repository A repository for developers to check in and check out business service artifacts. There are multiple check-in repositories. Each can be used for a different purpose (for example, development, production, testing, and so on).

connector Component-based interoperability model that enables third-party applications and JD Edwards EnterpriseOne to share logic and data. The JD Edwards EnterpriseOne connector architecture includes Java and COM connectors.

contra/clearing account A general ledger account in JD Edwards EnterpriseOne Financial Management that is used by the system to offset (balance) journal entries. For example, you can use a contra/clearing account to balance the entries created by allocations in JD Edwards EnterpriseOne Financial Management.

Control Table Workbench An application that, during the Installation Workbench processing, runs the batch applications for the planned merges that update the data dictionary, user-defined codes, menus, and user override tables.

control tables merge A process that blends a customer's modifications to the control tables with the data that accompanies a new release.

correlation data The data used to tie HTTP responses with requests that consist of business service name and method.

cost assignment The process in JD Edwards EnterpriseOne Advanced Cost Accounting of tracing or allocating resources to activities or cost objects.

cost component In JD Edwards EnterpriseOne Manufacturing, an element of an item's cost (for example, material, labor, or overhead).

credentials A valid set of JD Edwards EnterpriseOne username/password/environment/role, EnterpriseOne session, or EnterpriseOne token.

cross-reference utility services Utility services installed in a BPEL/ESB environment that access EnterpriseOne cross-reference data.

cross segment edit A logic statement that establishes the relationship between configured item segments. Cross segment edits are used to prevent ordering of configurations that cannot be produced.

currency restatement The process of converting amounts from one currency into another currency, generally for reporting purposes. You can use the currency restatement process, for example, when many currencies must be restated into a single currency for consolidated reporting.

cXML A protocol used to facilitate communication between business documents and procurement applications, and between e-commerce hubs and suppliers.

database credentials A valid database username/password.

database server A server in a local area network that maintains a database and performs searches for client computers.

Data Source Workbench An application that, during the Installation Workbench process, copies all data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the system-release number data source. It also updates the Data Source Plan detail record to reflect completion.

date pattern A calendar that represents the beginning date for the fiscal year and the ending date for each period in that year in standard and 52-period accounting.

denominated-in currency The company currency in which financial reports are based.

deployment artifacts Artifacts that are needed for the deployment process, such as servers, ports, and such.

deployment server A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

detail information Information that relates to individual lines in JD Edwards EnterpriseOne transactions (for example, voucher pay items and sales order detail lines).

direct connect A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one immediate and store-and-forward.

Do Not Translate (DNT) A type of data source that must exist on the iSeries because of BLOB restrictions.

dual pricing The process of providing prices for goods and services in two currencies.

duplicate published business services authorization records Two published business services authorization records with the same user identification information and published business services identification information.

embedded application server instance An OC4J instance started by and running wholly within JDeveloper.

edit code A code that indicates how a specific value for a report or a form should appear or be formatted. The default edit codes that pertain to reporting require particular attention because they account for a substantial amount of information.

edit mode A condition of a form that enables users to change data.

edit rule A method used for formatting and validating user entries against a predefined rule or set of rules.

Electronic Data Interchange (EDI) An interoperability model that enables paperless computer-to-computer exchange of business transactions between JD Edwards EnterpriseOne and third-party systems. Companies that use EDI must have translator software to convert data from the EDI standard format to the formats of their computer systems.

embedded event rule An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field based on a processing option value, and calling a business function. Contrast with the business function event rule.

Employee Work Center A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages.

enterprise server A server that contains the database and the logic for JD Edwards EnterpriseOne.

Enterprise Service Bus (ESB) Middleware infrastructure products or technologies based on web services standards that enable a service-oriented architecture using an event-driven and XML-based messaging framework (the bus).

EnterpriseOne administrator An actor responsible for the EnterpriseOne administration system.

EnterpriseOne credentials A user ID, password, environment, and role used to validate a user of EnterpriseOne.

EnterpriseOne object A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects.

EnterpriseOne development client Historically called "fat client," a collection of installed EnterpriseOne components required to develop EnterpriseOne artifacts, including the Microsoft Windows client and design tools.

EnterpriseOne extension A JDeveloper component (plug-in) specific to EnterpriseOne. A JDeveloper wizard is a specific example of an extension.

EnterpriseOne process A software process that enables JD Edwards EnterpriseOne clients and servers to handle processing requests and run transactions. A client runs one process, and servers can have multiple instances of a process. JD Edwards EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes don't have to wait if the server is particularly busy.

EnterpriseOne resource Any EnterpriseOne table, metadata, business function, dictionary information, or other information restricted to authorized users.

Environment Workbench An application that, during the Installation Workbench process, copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the system-release number data source. It also updates the Environment Plan detail record to reflect completion.

escalation monitor A batch process that monitors pending requests or activities and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time.

event rule A logic statement that instructs the system to perform one or more operations based on an activity that can occur in a specific application, such as entering a form or exiting a field.

explicit transaction Transaction used by a business service developer to explicitly control the type (auto or manual) and the scope of transaction boundaries within a business service.

exposed method or value object Published business service source files or parts of published business service source files that are part of the published interface. These are part of the contract with the customer.

facility An entity within a business for which you want to track costs. For example, a facility might be a warehouse location, job, project, work center, or branch/plant. A facility is sometimes referred to as a "business unit."

fast path A command prompt that enables the user to move quickly among menus and applications by using specific commands.

file server A server that stores files to be accessed by other computers on the network. Unlike a disk server, which appears to the user as a remote disk drive, a file server is a sophisticated device that not only stores files, but also manages them and maintains order as network users request files and make changes to these files.

final mode The report processing mode of a processing mode of a program that updates or creates data records.

foundation A framework that must be accessible for execution of business services at runtime. This includes, but is not limited to, the Java Connector and JDBj.

FTP server A server that responds to requests for files via file transfer protocol.

header information Information at the beginning of a table or form. Header information is used to identify or provide control information for the group of records that follows.

HTTP Adapter A generic set of services that are used to do the basic HTTP operations, such as GET, POST, PUT, DELETE, TRACE, HEAD, and OPTIONS with the provided URL.

instantiate A Java term meaning "to create." When a class is instantiated, a new instance is created.

integration developer The user of the system who develops, runs, and debugs the EnterpriseOne business services. The integration developer uses the EnterpriseOne business services to develop these components.

integration point (IP) The business logic in previous implementations of EnterpriseOne that exposes a document level interface. This type of logic used to be called XBPs. In EnterpriseOne 8.11, IPs are implemented in Web Services Gateway powered by webMethods.

integration server A server that facilitates interaction between diverse operating systems and applications across internal and external networked computer systems.

integrity test A process used to supplement a company's internal balancing procedures by locating and reporting balancing problems and data inconsistencies.

interface table See Z table.

internal method or value object Business service source files or parts of business service source files that are not part of the published interface. These could be private or protected methods. These could be value objects not used in published methods.

interoperability model A method for third-party systems to connect to or access JD Edwards EnterpriseOne.

in-your-face-error In JD Edwards EnterpriseOne, a form-level property which, when enabled, causes the text of application errors to appear on the form.

IServer service This internet server service resides on the web server and is used to speed up delivery of the Java class files from the database to the client.

jargon An alternative data dictionary item description that JD Edwards EnterpriseOne appears based on the product code of the current object.

Java application server A component-based server that resides in the middle-tier of a server-centric architecture. This server provides middleware services for security and state maintenance, along with data access and persistence.

JDBNET A database driver that enables heterogeneous servers to access each other's data.

JDEBASE Database Middleware A JD Edwards EnterpriseOne proprietary database middleware package that provides platform-independent APIs, along with client-to-server access.

JDECallObject An API used by business functions to invoke other business functions.

jde.ini A JD Edwards EnterpriseOne file (or member for iSeries) that provides the runtime settings required for JD Edwards EnterpriseOne initialization. Specific versions of the file or member must reside on every machine running JD Edwards EnterpriseOne. This includes workstations and servers.

JDEIPC Communications programming tools used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes.

jde.log The main diagnostic log file of JD Edwards EnterpriseOne. This file is always located in the root directory on the primary drive and contains status and error messages from the startup and operation of JD Edwards EnterpriseOne.

JDENET A JD Edwards EnterpriseOne proprietary communications middleware package. This package is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all JD Edwards EnterpriseOne supported platforms.

JDeveloper Project An artifact that JDeveloper uses to categorize and compile source files.

JDeveloper Workspace An artifact that JDeveloper uses to organize project files. It contains one or more project files.

JMS Queue A Java Messaging service queue used for point-to-point messaging.

listener service A listener that listens for XML messages over HTTP.

local repository A developer's local development environment that is used to store business service artifacts.

local standalone BPEL/ESB server A standalone BPEL/ESB server that is not installed within an application server.

Location Workbench An application that, during the Installation Workbench process, copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the system data source.

logic server A server in a distributed network that provides the business logic for an application program. In a typical configuration, pristine objects are replicated on to the logic server from the central server. The logic server, in conjunction with workstations, actually performs the processing required when JD Edwards EnterpriseOne software runs.

MailMerge Workbench An application that merges Microsoft Word 6.0 (or higher) word-processing documents with JD Edwards EnterpriseOne records to automatically print business documents. You can use MailMerge Workbench to print documents, such as form letters about verification of employment.

Manual Commit transaction A database connection where all database operations delay writing to the database until a call to commit is made.

master business function (MBF) An interactive master file that serves as a central location for adding, changing, and updating information in a database. Master business functions pass information between data entry forms and the appropriate tables. These master functions provide a common set of functions that contain all of the necessary default and editing rules for related programs. MBFs contain logic that ensures the integrity of adding, updating, and deleting information from databases.

master table See published table.

matching document A document associated with an original document to complete or change a transaction. For example, in JD Edwards EnterpriseOne Financial Management, a receipt is the matching document of an invoice, and a payment is the matching document of a voucher.

media storage object Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx.

message center A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user.

messaging adapter An interoperability model that enables third-party systems to connect to JD Edwards EnterpriseOne to exchange information through the use of messaging queues.

messaging server A server that handles messages that are sent for use by other programs using a messaging API. Messaging servers typically employ a middleware program to perform their functions.

Middle-Tier BPEL/ESB Server A BPEL/ESB server that is installed within an application server.

Monitoring Application An EnterpriseOne tool provided for an administrator to get statistical information for various EntepriseOne servers, reset statistics, and set notifications.

named event rule (NER)

Encapsulated, reusable business logic created using event rules, rather that C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

nota fiscal

In Brazil, a legal document that must accompany all commercial transactions for tax purposes and that must contain information required by tax regulations.

nota fiscal factura

In Brazil, a nota fiscal with invoice information.

See also *nota fiscal.*

Object Configuration Manager (OCM)

In JD Edwards EnterpriseOne, the object request broker and control center for the runtime environment. OCM keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, OCM directs access to it using defaults and overrides for a given environment and user.

Object Librarian

A repository of all versions, applications, and business functions reusable in building applications. Object Librarian provides check-out and check-in capabilities for developers, and it controls the creation, modification, and use of JD Edwards EnterpriseOne objects. Object Librarian supports multiple environments (such as production and development) and enables objects to be easily moved from one environment to another.

Object Librarian merge

A process that blends any modifications to the Object Librarian in a previous release into the Object Librarian in a new release.

Open Data Access (ODA)

An interoperability model that enables you to use SQL statements to extract JD Edwards EnterpriseOne data for summarization and report generation.

Output Stream Access (OSA)

An interoperability model that enables you to set up an interface for JD Edwards EnterpriseOne to pass data to another software package, such as Microsoft Excel, for processing.

package

JD Edwards EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the deployment server the installation program can find them. It is point-in-time snapshot of the central objects on the deployment server.

package build

A software application that facilitates the deployment of software changes and new applications to existing users. Additionally, in JD Edwards EnterpriseOne, a package build can be a compiled version of the software. When you upgrade your version of the ERP software, for example, you are said to take a package build.

Consider the following context: “Also, do not transfer business functions into the production path code until you are ready to deploy, because a global build of business functions done during a package build will automatically include the new functions.” The process of creating a package build is often referred to, as it is in this example, simply as “a package build.”

package location

The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path_code\package\package name. The subdirectories under this path are where the replicated objects for the package are placed. This is also referred to as where the package is built or stored.

Package Workbench

An application that, during the Installation Workbench process, transfers the package information tables from the Planner data source to the system-release number data source. It also updates the Package Plan detail record to reflect completion.

Pathcode Directory

The specific portion of the file system on the EnterpriseOne development client where EnterpriseOne development artifacts are stored.

patterns
General repeatable solutions to a commonly occurring problem in software design. For business service development, the focus is on the object relationships and interactions. For orchestrations, the focus is on the integration patterns (for example, synchronous and asynchronous request/response, publish, notify, and receive/reply).

planning family
A means of grouping end items whose similarity of design and manufacture facilitates being planned in aggregate.

preference profile
The ability to define default values for specified fields for a user-defined hierarchy of items, item groups, customers, and customer groups.

print server
The interface between a printer and a network that enables network clients to connect to the printer and send their print jobs to it. A print server can be a computer, separate hardware device, or even hardware that resides inside of the printer itself.

pristine environment
A JD Edwards EnterpriseOne environment used to test unaltered objects with JD Edwards EnterpriseOne demonstration data or for training classes. You must have this environment so that you can compare pristine objects that you modify.

processing option
A data structure that enables users to supply parameters that regulate the running of a batch program or report. For example, you can use processing options to specify default values for certain fields, to determine how information appears or is printed, to specify date ranges, to supply runtime values that regulate program execution, and so on.

production environment
A JD Edwards EnterpriseOne environment in which users operate EnterpriseOne software.

production-grade file server
A file server that has been quality assurance tested and commercialized and that is usually provided in conjunction with user support services.

Production Published Business Services Web Service
Published business services web service deployed to a production application server.

program temporary fix (PTF)
A representation of changes to JD Edwards EnterpriseOne software that your organization receives on magnetic tapes or disks.

project
In JD Edwards EnterpriseOne, a virtual container for objects being developed in Object Management Workbench.

promotion path
The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):

11>21>26>28>38>01

In this path, *11* equals new project pending review, *21* equals programming, *26* equals QA test/review, *28* equals QA test/review complete, *38* equals in production, *01* equals complete. During the normal project promotion cycle, developers check objects out of and into the development path code and then promote them to the prototype path code. The objects are then moved to the productions path code before declaring them complete.

proxy server
A server that acts as a barrier between a workstation and the internet so that the enterprise can ensure security, administrative control, and caching service.

published business service
EnterpriseOne service level logic and interface. A classification of a published business service indicating the intention to be exposed to external (non-EnterpriseOne) systems.

published business service identification information
Information about a published business service used to determine relevant authorization records. Published business services + method name, published business services, or *ALL.

published business service web service
Published business services components packaged as J2EE Web Service (namely, a J2EE EAR file that contains business service classes, business service foundation, configuration files, and web service artifacts).

published table
Also called a master table, this is the central copy to be replicated to other machines. Residing on the publisher machine, the F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

publisher
The server that is responsible for the published table. The F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

pull replication
One of the JD Edwards EnterpriseOne methods for replicating data to individual workstations. Such machines are set up as pull subscribers using JD Edwards EnterpriseOne data replication tools. The only time that pull subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the pull subscriber to the server machine that stores the F98DRPCN table.

QBE
An abbreviation for *query by example*. In JD Edwards EnterpriseOne, the QBE line is the top line on a detail area that is used for filtering data.

real-time event
A message triggered from EnterpriseOne application logic that is intended for external systems to consume.

refresh
A function used to modify JD Edwards EnterpriseOne software, or subset of it, such as a table or business data, so that it functions at a new release or cumulative update level, such as B73.2 or B73.2.1.

replication server
A server that is responsible for replicating central objects to client machines.

Rt-Addressing
Unique data identifying a browser session that initiates the business services call request host/port user session.

rules
Mandatory guidelines that are not enforced by tooling, but must be followed in order to accomplish the desired results and to meet specified standards.

quote order
In JD Edwards Procurement and Subcontract Management, a request from a supplier for item and price information from which you can create a purchase order.

In JD Edwards Sales Order Management, item and price information for a customer who has not yet committed to a sales order.

secure by default
A security model that assumes that a user does not have permission to execute an object unless there is a specific record indicating such permissions.

Secure Socket Layer (SSL)
A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

SEI implementation
A Java class that implements the methods that declare in a Service Endpoint Interface (SEI).

selection
Found on JD Edwards EnterpriseOne menus, a selection represents functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter.

serialize
The process of converting an object or data into a format for storage or transmission across a network connection link with the ability to reconstruct the original data or objects when needed.

Server Workbench
An application that, during the Installation Workbench process, copies the server configuration files from the Planner data source to the system-release number

data source. The application also updates the Server Plan detail record to reflect completion.

Service Endpoint Interface (SEI) A Java interface that declares the methods that a client can invoke on the service.

SOA Abbreviation for *Service Oriented Architecture*.

softcoding A coding technique that enables an administrator to manipulate site-specific variables that affect the execution of a given process.

source repository A repository for HTTP adapter and listener service development environment artifacts.

spot rate An exchange rate entered at the transaction level. This rate overrides the exchange rate that is set up between two currencies.

Specification merge A merge that comprises three merges: Object Librarian merge, Versions List merge, and Central Objects merge. The merges blend customer modifications with data that accompanies a new release.

specification A complete description of a JD Edwards EnterpriseOne object. Each object has its own specification, or name, which is used to build applications.

Specification Table Merge Workbench An application that, during the Installation Workbench process, runs the batch applications that update the specification tables.

SSL Certificate A special message signed by a certificate authority that contains the name of a user and that user's public key in such a way that anyone can "verify" that the message was signed by no one other than the certification authority and thereby develop trust in the user's public key.

store-and-forward The mode of processing that enables users who are disconnected from a server to enter transactions and then later connect to the server to upload those transactions.

subscriber table Table F98DRSUB, which is stored on the publisher server with the F98DRPUB table and identifies all of the subscriber machines for each published table.

superclass An inheritance concept of the Java language where a class is an instance of something, but is also more specific. "Tree" might be the superclass of "Oak" and "Elm," for example.

supplemental data Any type of information that is not maintained in a master file. Supplemental data is usually additional information about employees, applicants, requisitions, and jobs (such as an employee's job skills, degrees, or foreign languages spoken). You can track virtually any type of information that your organization needs.

For example, in addition to the data in the standard master tables (the Address Book Master, Customer Master, and Supplier Master tables), you can maintain other kinds of data in separate, generic databases. These generic databases enable a standard approach to entering and maintaining supplemental data across JD Edwards EnterpriseOne systems.

table access management (TAM) The JD Edwards EnterpriseOne component that handles the storage and retrieval of use-defined data. TAM stores information, such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions.

Table Conversion Workbench An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

table conversion An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

table event rules Logic that is attached to database triggers that runs whenever the action specified by the trigger occurs against the table. Although JD Edwards EnterpriseOne enables event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.

terminal server A server that enables terminals, microcomputers, and other devices to connect to a network or host computer or to devices attached to that particular computer.

three-tier processing The task of entering, reviewing and approving, and posting batches of transactions in JD Edwards EnterpriseOne.

three-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing receipt information to supplier's invoices to create vouchers. In a three-way match, you use the receipt records to create vouchers.

transaction processing (TP) monitor A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and may include programs that validate data and format terminal screens.

transaction processing method A method related to the management of a manual commit transaction boundary (for example, start, commit, rollback, and cancel).

transaction set An electronic business transaction (electronic data interchange standard document) made up of segments.

trigger One of several events specific to data dictionary items. You can attach logic to a data dictionary item that the system processes automatically when the event occurs.

triggering event A specific workflow event that requires special action or has defined consequences or resulting actions.

two-way authentication An authentication mechanism in which both client and server authenticate themselves by providing the SSL certificates to each other.

two-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing purchase order detail lines to the suppliers' invoices to create vouchers. You do not record receipt information.

user identification information User ID, role, or *public.

User Overrides merge Adds new user override records into a customer's user override table.

value object A specific type of source file that holds input or output data, much like a data structure passes data. Value objects can be exposed (used in a published business service) or internal, and input or output. They are comprised of simple and complex elements and accessories to those elements.

variance In JD Edwards Capital Asset Management, the difference between revenue generated by a piece of equipment and costs incurred by the equipment.

In JD Edwards EnterpriseOne Project Costing and JD Edwards EnterpriseOne Manufacturing, the difference between two methods of costing the same item (for example, the difference between the frozen standard cost and the current cost is an engineering variance). Frozen standard costs come from the Cost Components table, and the current costs are calculated using the current bill of material, routing, and overhead rates.

versioning a published business service Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.

Version List merge The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release, as well as their processing options data.

visual assist Forms that can be invoked from a control via a trigger to assist the user in determining what data belongs in the control.

vocabulary override An alternate description for a data dictionary item that appears on a specific JD Edwards EnterpriseOne form or report.

wchar_t An internal type of a wide character. It is used for writing portable programs for international markets.

web application server A web server that enables web applications to exchange data with the back-end systems and databases used in eBusiness transactions.

web server A server that sends information as requested by a browser, using the TCP/IP set of protocols. A web server can do more than just coordination of requests from browsers; it can do anything a normal server can do, such as house applications or data. Any computer can be turned into a web server by installing server software and connecting the machine to the internet.

Web Service Description Language (WSDL) An XML format for describing network services.

Web Service Inspection Language (WSIL) An XML format for assisting in the inspection of a site for available services and a set of rules for how inspection-related information should be made.

web service proxy foundation Foundation classes for web service proxy that must be included in a business service server artifact for web service consumption on WAS.

web service softcoding record An XML document that contains values that are used to configure a web service proxy. This document identifies the endpoint and conditionally includes security information.

web service softcoding template An XML document that provides the structure for a soft coded record.

Where clause The portion of a database operation that specifies which records the database operation will affect.

Windows terminal server A multiuser server that enables terminals and minimally configured computers to display Windows applications even if they are not capable of running Windows software themselves. All client processing is performed centrally at the Windows terminal server and only display, keystroke, and mouse commands are transmitted over the network to the client terminal device.

wizard A type of JDeveloper extension used to walk the user through a series of steps.

workbench A program that enables users to access a group of related programs from a single entry point. Typically, the programs that you access from a workbench are used to complete a large business process. For example, you use the JD Edwards EnterpriseOne Payroll Cycle Workbench (P07210) to access all of the programs that the system uses to process payroll, print payments, create payroll reports, create journal entries, and update payroll history. Examples of JD Edwards EnterpriseOne workbenches include Service Management Workbench (P90CD020), Line Scheduling Workbench (P3153), Planning Workbench (P13700), Auditor's Workbench (P09E115), and Payroll Cycle Workbench.

work day calendar In JD Edwards EnterpriseOne Manufacturing, a calendar that is used in planning functions that consecutively lists only working days so that component and work order scheduling can be done based on the actual number of work days available. A work

day calendar is sometimes referred to as planning calendar, manufacturing calendar, or shop floor calendar.

workflow The automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

workgroup server A server that usually contains subsets of data replicated from a master network server. A workgroup server does not perform application or batch processing.

XAPI events A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and then calls third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when the specified transactions occur to return a response.

XML CallObject An interoperability capability that enables you to call business functions.

XML Dispatch An interoperability capability that provides a single point of entry for all XML documents coming into JD Edwards EnterpriseOne for responses.

XML List An interoperability capability that enables you to request and receive JD Edwards EnterpriseOne database information in chunks.

XML Service An interoperability capability that enables you to request events from one JD Edwards EnterpriseOne system and receive a response from another JD Edwards EnterpriseOne system.

XML Transaction An interoperability capability that enables you to use a predefined transaction type to send information to or request information from JD Edwards EnterpriseOne. XML transaction uses interface table functionality.

XML Transaction Service (XTS) Transforms an XML document that is not in the JD Edwards EnterpriseOne format into an XML document that can be processed by JD Edwards EnterpriseOne. XTS then transforms the response back to the request originator XML format.

Z event A service that uses interface table functionality to capture JD Edwards EnterpriseOne transactions and provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested to be notified when certain transactions occur.

Z table A working table where non-JD Edwards EnterpriseOne information can be stored and then processed into JD Edwards EnterpriseOne. Z tables also can be used to retrieve JD Edwards EnterpriseOne data. Z tables are also known as interface tables.

Z transaction Third-party data that is properly formatted in interface tables for updating to the JD Edwards EnterpriseOne database.

Index

C

D

E

F

G

I

K

L

M

N

P

R

S

T

U

V

W

X