

SeeBeyond ICAN Suite

e*Way Intelligent Adapter for Oracle Manufacturing User's Guide

Release 5.0.5 for Schema Run-time Environment (SRE)

Java Version



The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e*Gate, e*Way, and e*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2005 SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20050406023638.

Contents

Preface	11
Intended Reader	11
Organization	11
Nomenclature	12
Online Use	12
Writing Conventions	12
Additional Documentation	13

Chapter 1

Introduction	14
Oracle Manufacturing	14
Basic Operation	15
Oracle Manufacturing e*Way	16
Basic Operation	17
Disclaimer	18
e*Way Components	18
Supported Operating Systems	18
Supported External Systems	19

Chapter 2

Installation	20
System Requirements	20
External System Requirements	21
GUI Host	21
Participating Host	21
Oracle Patches	21
Environment Configuration	21
Installing the e*Way	22

Contents

Installation Procedure	22
Windows Systems	22
UNIX Systems	26
Subdirectories and Files	27
Registering the DataSource	30
ODBC Test	31
Optional Example Files	31
Installation Procedure	32
Subdirectories and Files	33

Chapter 3

System Implementation	35
Oracle Setup	35
Oracle Implementation Notes	37
e*Gate Setup	37
Implementation Sequence	38
Viewing e*Gate Components	38
Creating a Schema	39
Creating Event Types	40
Generating Event Type Definitions	41
Using the ETD Builder	41
Assigning ETDs to Event Types	46
Creating Intelligent Queues	47
Defining Collaborations	48
The Java Collaboration Rules Editor	48
Subcollaboration Rules	48
Running the e*Way	49
Configuring the Schema	49
Run the Schema	49
To start the Control Broker:	49
Sample Schemas	51
POImport	51
POReceiving	52

Chapter 4

Pre-Validation Process	53
Overview	53
Pre-Validation Scripts	54
Disclaimer	54
Pre-Validation Procedures	54

Common Procedures	54
Interface-Specific Procedures	54
Inventory Module	55
Customer Items	55
Oracle Interface Tables	55
SeeBeyond Staging Tables	55
Pre-Validation Scripts	55
Customer Items Cross-Reference	56
Oracle Interface Tables	56
SeeBeyond Staging Tables	56
Pre-Validation Scripts	56
Cycle Count Entries	57
Oracle Interface Table	57
SeeBeyond Staging Table	57
Pre-Validation Scripts	57
Items and Item Revisions	58
Oracle Interface Tables	58
SeeBeyond Staging Tables	58
Pre-Validation Scripts	58
Replenishment	59
Oracle Interface Tables	59
SeeBeyond Staging Tables	59
Pre-Validation Scripts	59
Transactions	60
Oracle Interface Tables	60
SeeBeyond Staging Tables	60
Pre-Validation Scripts	60
Purchase Order Import Module	61
Purchase Order Import	61
Oracle Interface Tables	61
SeeBeyond Staging Tables	61
Pre-Validation Scripts	61
Purchase Order Receiving	62
Oracle Interface Tables	62
SeeBeyond Staging Tables	62
Pre-Validation Scripts	62
Purchase Order Requisitions	63
Oracle Interface Table	63
SeeBeyond Staging Table	63
Pre-Validation Scripts	63
Building a Custom Pre-Validation Package	64
Concurrent Manager Request Function	64
Template DTD	65
Naming Conventions	67
Oracle Interface Name	67
Short Table Name	67
Stored Procedures	67
Staging Table Node	67
SQL File Name	67
Concurrent Manager Request Function	67

Chapter 5

e*Way Setup	68
Overview	68
Setting Up the e*Way	69
Creating the e*Way	69
Modifying e*Way Properties	70
Configuring the e*Way	71
Changing the User Name	72
Setting Startup Options or Schedules	72
Activating or Modifying Logging Options	74
Activating or Modifying Monitoring Thresholds	75
Schema Designer Creating e*Way Connections	76
Managing e*Way Connections	79
Controlling When a Connection is Made	79
Controlling When a Connection is Disconnected	80
Controlling the Connectivity Status	80
Using the e*Way Configuration Editor	81
Section and Parameter Controls	82
Parameter Configuration Controls	82
File Menu Selections	83
Command-line Configuration	83
Getting Help	83
Troubleshooting the e*Way	84
Configuration Problems	84
System-related Problems	85

Chapter 6

Operational Overview	86
Interaction With Oracle	86
Generating an ETD	86
Exposed ETD Nodes	88
Staging Table Node	88
COUNT	88
DELETE	89
INITIALIZE	90
MOVE	91
REQUEST	92
REQUEST_STATUS	92
VALIDATE	93
Transmitting Data	94
Basic e*Gate/e*Way Operation	96
Event Type Definitions	96
ETD Components	96
ETD Builders	97
Collaborations	98
Java Collaboration Service	99

e*Way Connections	100
Establishing Connections	100

Chapter 7

Configuration Parameters	102
Overview	102
e*Way Connections	102
e*Way Connections	103
DataSource Settings	103
class	103
DriverType	103
ServerName	104
PortNumber	104
DatabaseName	104
user name	104
password	104
timeout	105
Connector Settings	106
type	106
class	106
transaction mode	106
connection establishment mode	107
connection inactivity timeout	107
connection verification interval	107

Chapter 8

Java Classes and Methods	108
Overview	108
Object Classes	108
StatementAgent Class	109
Methods	110
cancel	110
clearBatch	110
clearWarnings	111
executeBatch	111
getFetchDirection	112
getFetchSize	112
getMaxFieldSize	113
getMaxRows	113
getMoreResults	114
getQueryTimeout	114
getResultSet	115
getResultSetConcurrency	115
getResultSetType	116
getUpdateCount	116
getWarnings	117
isClosed	117
queryDescription	118
queryName	118
resetRequested	119
resultSetConcurToString	119
resultSetDirToString	120

resultSetTypeToString	120
sessionClosed	121
sessionOpen	121
setCursorName	122
setEscapeProcessing	122
setFetchDirection	123
setMaxRows	123
setQueryTimeout	124
setMaxFieldSize	124
stmtInvoke	125
PreparedStatementAgent Class	126
Methods	127
addBatch	127
clearParameters	127
execute	127
executeQuery	128
executeUpdate	128
sessionOpen	129
setNull	129
setNull	130
setObject	130
setObject	131
setObject	131
setBoolean	132
setByte	132
setShort	133
setInt	133
setLong	134
setFloat	134
setDouble	135
setBigDecimal	135
setDate	136
setDate	136
setTime	137
setTime	137
setTimestamp	138
setTimestamp	138
setString	139
setBytes	139
setAsciiStream	140
setBinaryStream	140
setCharacterStream	141
setArray	142
setBlob	142
setClob	143
setRef	143
CallableStatementAgent Class	144
Constructors	145
CallableStatementAgent	145
CallableStatementAgent	145
CallableStatementAgent	146
Methods	147
sessionOpen	147
registerOutParameter	147
registerOutParameter	148
registerOutParameter	148
wasNull	149
getObject	149
getObject	150
getBoolean	150
getByte	151
getShort	151

getInt	152
getLong	152
getFloat	153
getDouble	153
getBigDecimal	154
getDate	154
getDate	155
getTime	155
getTime	156
getTimestamp	156
getTimestamp	157
getString	157
getBytes	158
getArray	158
getBlob	159
getClob	159
getRef	160
SqlStatementAgent Class	161
Constructors	162
SqlStatementAgent	162
SqlStatementAgent	162
Methods	163
addBatch	163
execute	163
executeQuery	164
executeUpdate	164

Appendix A

Validation Scripts 165

Inventory	165
Customer Items	165
Customer Item Cross-Reference	168
Cycle Count Entries	170
Items	172
Item Revisions	178
Replenishment	179
Master	179
Headers	179
Lines	181
Transactions	183
Lots	186
Serial Numbers	186
Purchase Order	187
Import	187
Master	187
Headers	187
Lines	191
Receiving	196
Master	196
Header	196
Transactions	199
Requisitions	206

Appendix B

Error Codes 213

Inventory	213
Customer Items	213
Customer Item Cross-Reference	215
Cycle Count Entries	217
Items	219
Item Revisions	223
Replenishment	224
Master	224
Headers	224
Lines	225
Transactions	227
Lots	229
Serial Numbers	229
Purchase Order	230
Import	230
Master	230
Headers	230
Lines	233
Receiving	237
Master	237
Header	237
Transactions	239
Requisitions	244

Index 249

Preface

This Preface contains information regarding the User's Guide itself.

P.1 Intended Reader

The reader of this guide is presumed to be a developer or system administrator with responsibility for maintaining the SeeBeyond™ e*Gate™ Integrator system, and have a working knowledge of:

- Operation and administration of Windows 2000, and/or UNIX systems
- Windows-style GUI operations
- Oracle Manufacturing concepts and operations
- Integrating Oracle Manufacturing applications with external systems

P.2 Organization

This User's Guide is organized roughly into two parts. The first part, consisting of Chapters 1-5, introduces the e*Way™ and describes the procedures for installing and setting up the e*Way, and implementing a working system incorporating the e*Way. This part should be of particular interest to a System Administrator or other user charged with the task of getting the system up and running.

The second part, consisting of Chapters 6-8, describes the details of e*Way operation and configuration, including descriptions of the exposed Java methods. This part should be of particular interest to a Developer involved in customizing the e*Way for a specific purpose. Information contained in this part that is necessary for the initial setup of the e*Way is cross-referenced in the first part of the guide, at the appropriate points in the procedures.

A comprehensive listing of the pre-validation scripts and error codes are included as Appendix A and B, respectively.

P.3 Nomenclature

Note that for purposes of brevity, the e*Way Intelligent Adapter for Oracle Manufacturing is frequently referred to as the Oracle Manufacturing e*Way, or simply the e*Way.

P.4 Online Use

This User's Guide is provided in Adobe Acrobat's Portable Document Format (PDF). As such, it can be printed out on any printer or viewed online. When viewing online, you can take advantage of the extensive hyperlinking imbedded in the document to navigate quickly throughout the Guide.

Hyperlinking is available in:

- The Table of Contents
- The Index
- Within the chapter text, indicated by **blue print**

Existence of a hyperlink *hotspot* is indicated when the hand cursor points to the text. Note that the hotspots in the Index are the *page numbers*, not the topics themselves. Returning to the spot you hyperlinked from is accomplished by right-clicking the mouse and selecting **Go To Previous View** on the resulting menu.

P.5 Writing Conventions

The writing conventions listed in this section are observed throughout this document.

Monospaced (Courier) Font

Computer code and text to be typed at the command line are set in Courier as shown below.

```
Configuration for BOB_Promotion
java -jar ValidationBuilder.jar
```

Variables within a command line, or attributes within a method signature, are set in italics as shown below:

```
stcregutl -rh host-name -un user-name -up password -sf
```

Bold Sans-serif Font

- User Input: Click **Apply** to save, or **OK** to save and close.
- File Names and Paths: In the **Open** field, type **D:\setup\setup.exe**.
- Parameter, Function, and Command Names: The default parameter **localhost** is usually used only for testing.

P.6 Additional Documentation

- Many of the procedures included in this User's Guide are described in greater detail in the *e*Gate Integrator User's Guide*
- For more information on the Java Collaboration Service, see the *e*Gate Integrator Collaboration Services Reference*

Introduction

This chapter provides a brief introduction to the SeeBeyond e*Way Intelligent Adapter for Oracle Manufacturing.

1.1 Oracle Manufacturing

The Oracle E-Business Suite 11i is a comprehensive enterprise resource planning (ERP) software package built upon Oracle's database technology. It is presented within an Internet environment, using online transaction processing, to address the global requirements of today's typical enterprise.

The E-Business suite includes a large number of Product Families, grouped into software modules corresponding to what were once stand-alone computer systems used by individual departments. These Product Families are identified by their major business functions, such as:

- Financials
- Human Resources
- Manufacturing
- Marketing
- Sales

These Product Families are integrated together to share a common database, allowing a company's various departments to quickly and easily share information and communicate with each other.

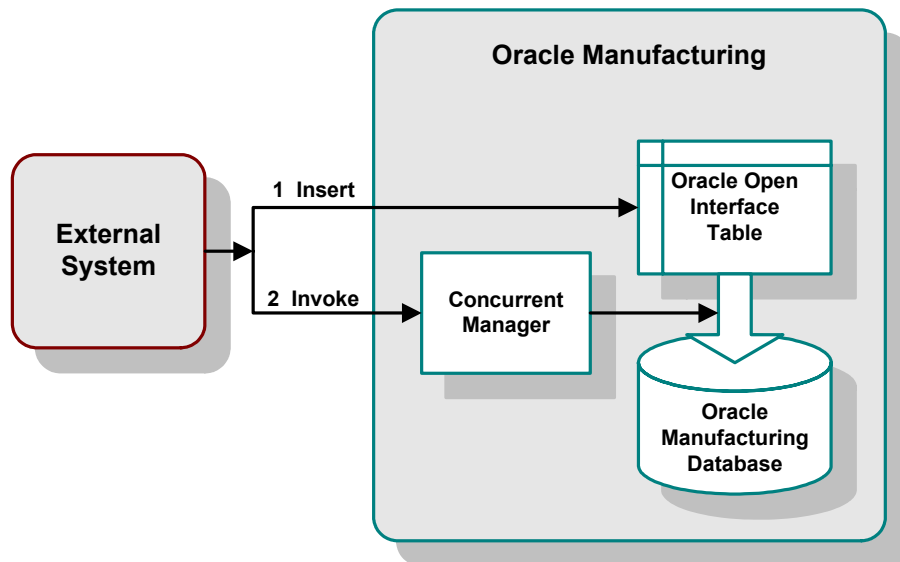
One of these Product Families, Oracle Manufacturing, contains modules designed to manage the production process. Oracle Manufacturing is closely integrated with Oracle Financials and Oracle's Supply-Chain Product Families to provide real-time collaborations with suppliers, subcontractors, and logistics partners. Oracle Manufacturing includes an extensive number of modules, including:

- Bills of Material
- Cost Management
- Inventory Management
- Production Management
- Purchasing

1.1.1 Basic Operation

The basic architecture of an Oracle system contains a set of base objects, which are held in highly normalized core tables within the Oracle database. A de-normalized view of these base objects is provided in a set of Open Interface Tables (OITs), also maintained in the database. Data is passed from the Open Interface Tables to the core tables under the control of the Concurrent Manager.

Figure 1 Basic Oracle Manufacturing Operation



In a typical scenario, an operator schedules an import job by means of the Oracle front end, which initiates the following procedure:

- 1 Data is passed from the Open Interface Tables to the core tables under the control of Import Jobs schedule by the Concurrent Manager.
- 2 It then invokes the Oracle Concurrent Manager, which:
 - A Validates the data in the Open Interface Table, based on a set of stored SQL procedures.
 - B Inserts the validated rows into the Oracle Manufacturing Database.

This highly-simplified procedure is depicted in Figure 1. There are several limitations to this very basic scheme:

- Once data is in the Open Interface Table, it cannot be withdrawn or corrected
- Data failing the validation process may be handled in different ways—some import scripts update the original rows with error codes, while other scripts log errors to a file, requiring user intervention
- Only the default validation rules provided by Oracle are used in the validation process, and may not address specific customer requirements
- There is no easy way to insert batches of data as a transactional unit—for example, where all inserts from a batch must succeed (if *any* fail, then *all* must fail)

1.2 Oracle Manufacturing e*Way

The Oracle Manufacturing e*Way provides a comprehensive integration solution for the Oracle Manufacturing Product Family included in Oracle release 11i. It provides added value to the suite by allowing pre-validation of data before it is uploaded to the Oracle database, and providing the ability to handle error conditions. These are important features, since data which is even slightly erroneous will cause a transmission to fail at runtime.

Oracle contains three types of Open Interfaces:

- API (PL/SQL)
- View
- Tables

The API and View Open Interfaces can be managed using the SeeBeyond Oracle e*Way, while the Oracle Manufacturing e*Way provides additional support for Open Interface Tables. Additionally, the e*Way is highly customizable—allowing you to modify the validation rules or add your own customer-specific requirements for error handling and recovery. For example, you might choose to have records that fail an import remain in the OIT so that an operator will be able to use the Oracle administration tools to amend and resubmit the data. Alternatively, you could have errors sent back through e*Gate to the calling application—or select a mixture of both, depending on the type of message.

Using the e*Way, you can insert and update Oracle Manufacturing with data from external systems using the e*Gate Schema Designer. Information can be sent either in real-time, delivering prompt event-driven performance, or in batch (store and forward) form. Error handling is provided through extensive pre-validation business logic. The following Oracle modules and Open Interfaces are currently supported:

- Purchasing
 - ♦ Purchase Order Import
 - ♦ Purchase Order Receiving
 - ♦ Purchase Order Requisition
- Inventory Management
 - ♦ Customer Items Interface
 - ♦ Customer Items Cross-Reference Interface
 - ♦ Cycle Count Entries Interface
 - ♦ Item Import
 - ♦ Item Revisions Import
 - ♦ Replenishment Interface
 - ♦ Transactions Interface

The e*Way includes an Event Type Definition (ETD) Wizard to generate Java ETDs specifically tailored to each Oracle Manufacturing module. ETDs contain the formats for Events (packets of data) that are processed by the e*Gate Integrator system. The Wizard creates ETDs that describe the hierarchies of the interface tables for the particular module. These ETDs also support the grouping of input Events. See [Generating an ETD](#) on page 86 for information on the functionality of the ETD Wizard.

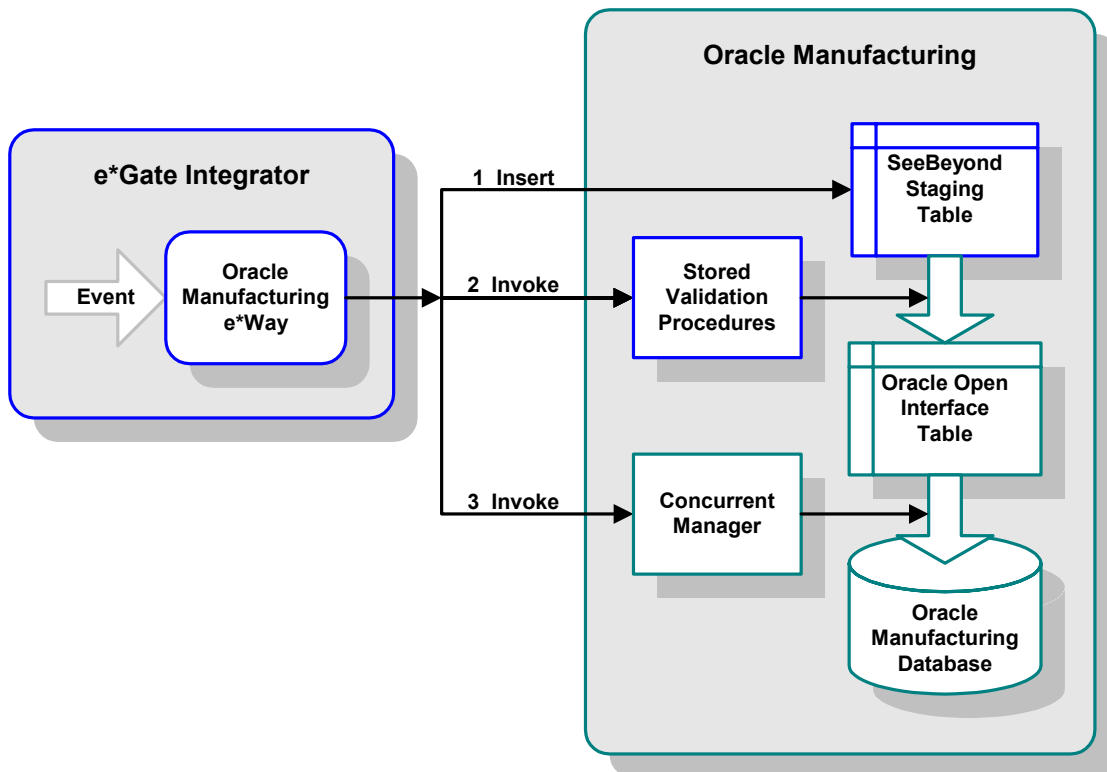
1.2.1 Basic Operation

When the Oracle Manufacturing e*Way receives an Event via the e*Gate Integrator system, it proceeds as follows:

- 1 It inserts the data into a staging table contained within the Oracle database, following the ETD corresponding to the current Open Interface Table.
- 2 It pre-validates the data in the staging table by invoking stored procedures, also contained in the Oracle database. The e*Way checks for and manages any errors, after which the validated rows are placed into the Interface Table.
- 3 It then initiates a standard Oracle validation by invoking the Oracle Concurrent Manager, which validates the data in the Open Interface Table and stores the validated rows in the Oracle Manufacturing Database.

This simplified procedure is depicted in Figure 2. A more detailed description, including the handling of data failing validation, is contained in [Chapter 6](#).

Figure 2 e*Gate-to-Oracle Operation



1.2.2 Disclaimer

The pre-validation scripts packaged with the e*Way for the supported modules are rudimentary and do not cover all possible scenarios. Since the requirements from Oracle constantly change and evolve, you may be required to provide additional data and/or parameters to ensure that the Concurrent Manager request completes successfully, without errors.

1.3 e*Way Components

The Oracle Manufacturing e*Way incorporates the following components:

- SeeBeyond Database e*Way (installed with e*Gate)
- Database e*Way Connections, which allow configurable connection to a specified external system
- Default Configuration Definition file
 - ♦ `oracle.def`
- Java classes, containing the logic required by the e*Way to interact with the external databases
 - ♦ `stcjdbcx.jar`
- The Oracle Manufacturing ETD Builder/Wizard
- Supporting methods and Event Type Definitions

For a list of installed files, see [Installing the e*Way](#) on page 22.

1.4 Supported Operating Systems

The e*Way Intelligent Adapter for Oracle Manufacturing currently supports the following operating systems.

Note: The e*Gate Schema Designer runs only on Windows.

- Windows 2000 and Windows Server 2003
- HP-UX 11.0 and 11i (PA-RISC)
- Sun Solaris 8

1.5 Supported External Systems

The e*Way Intelligent Adapter for Oracle Manufacturing currently supports the following Oracle system components:

- Oracle Applications 11i (11.5.3)

Installation

This chapter describes the requirements and procedures for installing the SeeBeyond e*Way Intelligent Adapter for Oracle Manufacturing. Following installation, you must configure it for your system (see [Chapter 7](#)) and incorporate it into a schema (see [Chapter 3](#)).

Note: Please read the *readme.txt* file located in the *addons\ewom* directory on the installation CD-ROM for important information regarding this installation.

2.1 System Requirements

To use the Oracle Manufacturing e*Way, you need the following:

- 1 An e*Gate Participating Host.
- 2 A TCP/IP network connection.
- 3 Sufficient free disk space to accommodate e*Way files (not including sample schemas):
 - ♦ Approximately 2.8 MB on Windows systems
 - ♦ Approximately 3.9 MB on Solaris systems
 - ♦ Approximately 3.8 MB on HP-UX systems
 - ♦ Approximately 4.0 MB s on AIX systems

Additional disk space is required to process and queue the data that this e*Way processes; the amount necessary varies, based on the type and size of the data being processed.

2.2 External System Requirements

2.2.1 GUI Host

The e*Gate Schema Designer GUI runs only on the Windows 2000 operating systems. To enable the e*Gate editors to communicate with the external system, the following items must be installed on any platforms running the e*Gate GUI.

- The Oracle Client Library must be installed on the GUI host to utilize the ETD Builder.
 - ♦ Oracle Applications 11i (11.5.3)
- Microsoft Data Access Components (MDAC) RTM version 2.6 or greater (included in the e*Gate installation)
- Oracle OLE DB Version 8.1.7.2.1 or later. This driver is needed by the e*Gate ETD Editor GUI for accurate identification of stored procedures during the generation of database ETDs. You can download this driver from the Oracle OTN website:

http://www.technet.oracle.com/software/tech/nt/ole_db/software_index.htm

2.2.2 Participating Host

A Participating Host is the platform used to run the e*Gate schema, and, depending upon the operating system, may or may not also serve as the e*Gate GUI host.

The e*Way installation program installs the Type 4 JDBC drivers required to connect to the external Oracle database. These drivers are correct for most Oracle implementations. For implementations using protocols other than Type 4, however, the appropriate client drivers must be installed. These client drivers can be found on the Oracle client installation disc.

A complete list of drivers from third party vendors is also available at the following URL:

<http://industry.java.sun.com/products/jdbc/drivers>

2.2.3 Oracle Patches

Patch 1312573 (see Oracle doc ID 110328.1) is required to fix a Create Release problem that occurs with database release 11.5.4 and earlier.

2.3 Environment Configuration

No changes are required to the Participating Host's operating environment to support this e*Way.

2.4 Installing the e*Way

2.4.1 Installation Procedure

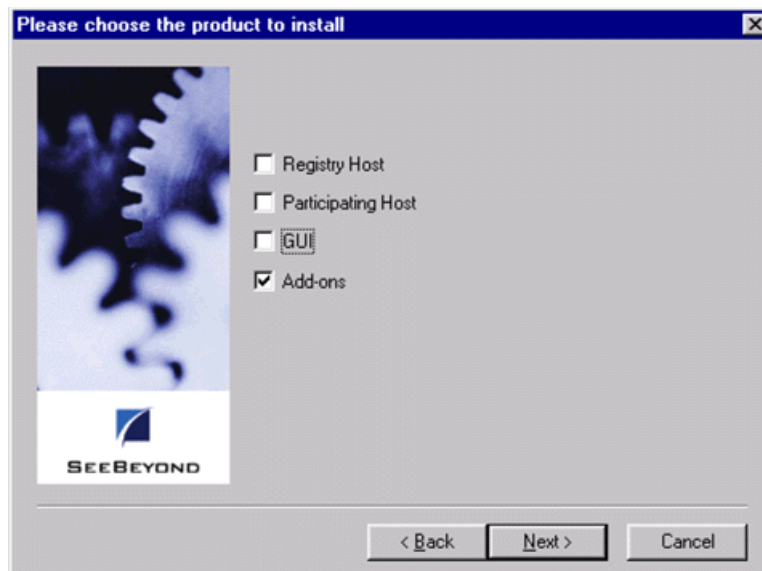
Windows Systems

Note: *The installation utility detects and suggests the appropriate installation directory. Use this directory unless advised otherwise by SeeBeyond. You must have Administrator privileges to install this e*Way.*

To Install the e*Way on a Windows 2000 System

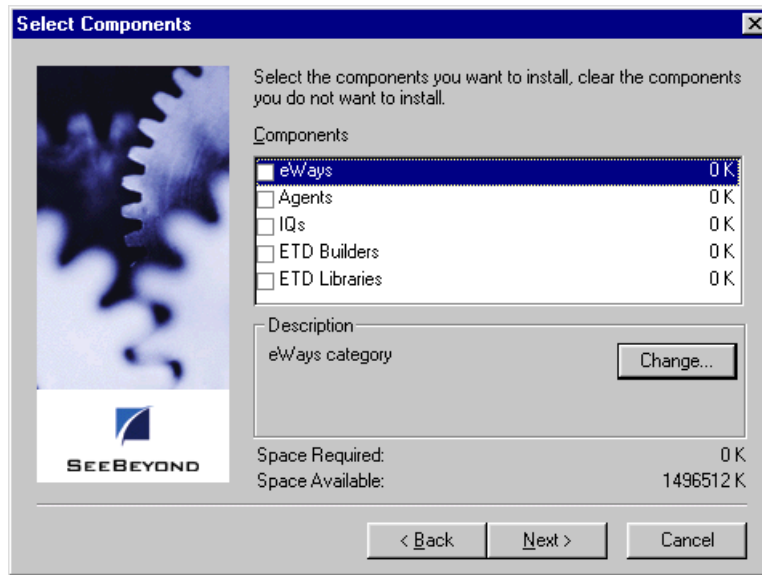
- 1 Log in as an Administrator on the workstation on which you want to install the e*Way.
- 2 Exit all Windows programs and disable any anti-virus applications before running the setup program.
- 3 Insert the e*Way installation CD-ROM into the CD-ROM drive.
- 4 If the CD-ROM drive's Autorun feature is enabled, the setup application should launch automatically. Otherwise, use the Windows Explorer or the Control Panel's **Add/Remove Applications** feature to launch the file **setup.exe** on the CD-ROM drive.
- 5 The InstallShield setup application launches. Follow the on-screen instructions until you come to the **Choose Product** screen.

Figure 3 Choose Product Dialog



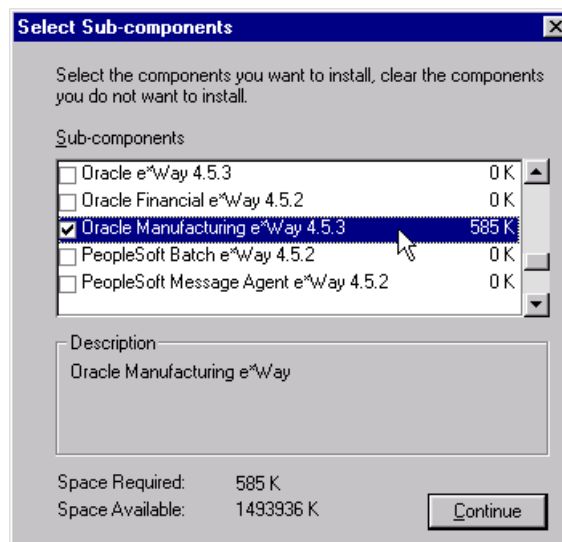
- 6 Check **Add-ons**, then click **Next**. Again follow the on-screen instructions.

Figure 4 Select Components Dialog (1)



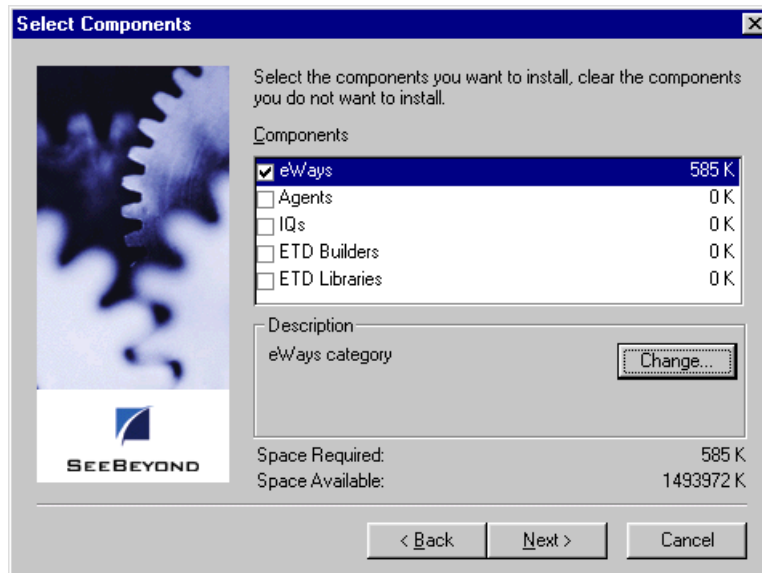
- 7 When the **Select Components** dialog box appears, highlight—but do not check—**eWays**, and then click **Change**.

Figure 5 Select Sub-components Dialog



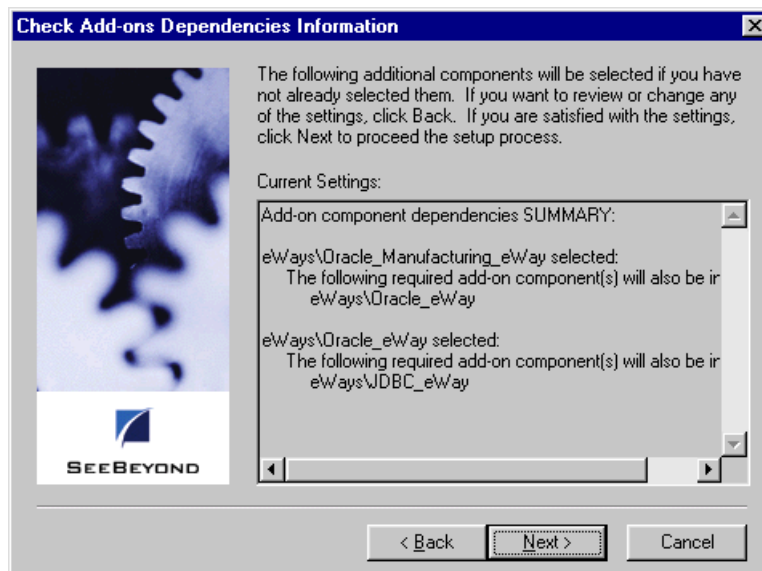
- 8 When the **Select Sub-components** dialog box appears, check **Oracle Manufacturing eWay**.
- 9 Click **Continue**, and the **Select Components** dialog box reappears, showing your selection.

Figure 6 Select Components Dialog (2)



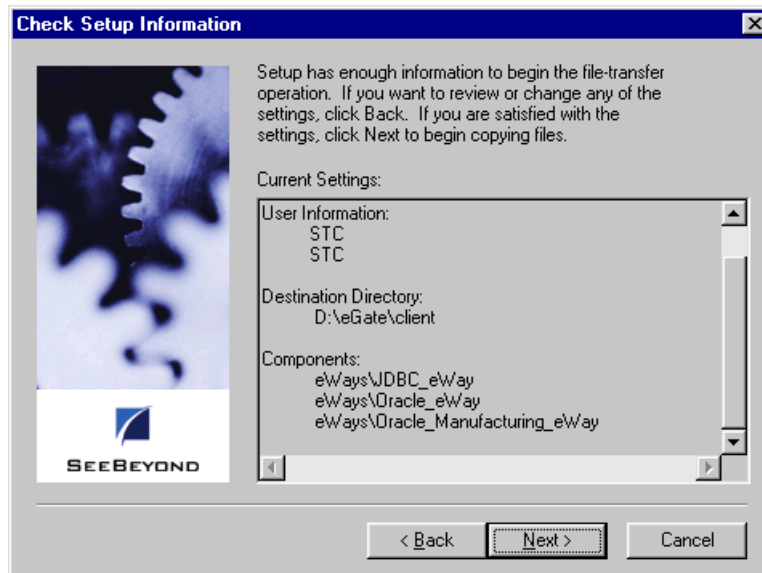
- 10 At this point, the **Space Required** figure only includes part of the total, since there are dependency e*Ways involved in this installation.
- 11 Click **Next** and the **Check Add-ons Dependencies Information** dialog appears. Here you see that both the Oracle e*Way and JDBC e*Way will be installed as part of the Oracle Manufacturing e*Way installation.

Figure 7 Check Add-ons Dependencies Information Dialog



- 12 Click **Next** and the **Check Setup Information** dialog appears, summarizing the setup information.

Figure 8 Check Setup Information Dialog



- 13 Click **Next** and the installation process begins. At the conclusion of the installation, the **Setup Complete** dialog appears, confirming correct installation.

Figure 9 Setup Complete Dialog



- 14 Click **Finish** to close the setup program.

UNIX Systems

Note: *You are not required to have root privileges to install this e*Way. Log on under the user name that you wish to own the e*Way files. Be sure that this user has sufficient privilege to create files in the e*Gate directory tree.*

To Install the e*Way on a UNIX System

- 1 Log onto the workstation containing the CD-ROM drive and, if necessary, mount the drive.
- 2 Insert the e*Way installation CD-ROM into the CD-ROM drive.
- 3 At the shell prompt, type
`cd /cdrom`
- 4 Start the installation script by typing:
`setup.sh`
- 5 A menu of options appears. Select the **Install e*Way** option and follow any additional on-screen instructions.

Note: *The installation utility detects and suggests the appropriate installation directory. Use this directory unless advised otherwise by SeeBeyond. Note also that **no spaces** should appear in the installation path name.*

2.4.2 Subdirectories and Files

Note: *Installing the e*Way Intelligent Adapter for Oracle Manufacturing installs both Java and Monk versions of the Oracle e*Way. Only the files used by the Java version are listed in this section.*

By default, the InstallShield installer creates the following subdirectories and installs the following files within the \eGate\client tree on the Participating Host, and the \eGate\Server\registry\repository\default tree on the Registry Host.

Table 1 Participating Host & Registry Host

Subdirectories	Files
\bin\	jcscomp.jar stcjintegra.jar xerces.jar
\Classes\	stjcs.jar stcjdbcx.jar
\configs\ewOracleApps\ewom\Inventory\ Customer_Item\	Customer_Item.xml
\configs\ewOracleApps\ewom\Inventory\ Customer_Item_Xref\	Customer_Item_Xref.xml
\configs\ewOracleApps\ewom\Inventory\ Cycle_Count\	Cycle_Count.xml
\configs\ewOracleApps\ewom\Inventory\ Item_Import\	Item_Import.xml
\configs\ewOracleApps\ewom\Inventory\ Replenish\	Replenishment_Import.xml
\configs\ewOracleApps\ewom\Inventory\ Transaction\	Item_Transactions.xml
\configs\ewOracleApps\ewom\Purchase\ Order_Import\	Order_Import.xml
\configs\ewOracleApps\ewom\Purchase\ Order_Receiving\	Order_Receiving.xml
\configs\ewOracleApps\ewom\Purchase\ Order_Requisition\	Order_Requisition.xml
\configs\ewOracleApps\template\	templateXml.xml
\configs\jdbcodbc\	jdbcodbc.def
\configs\oracle\	oracle.def
\etd\	jdbcodbc.ctl oracle.ctl dbwizard.ctl

Table 1 Participating Host & Registry Host

Subdirectories	Files
\etd\db\	Com_stc_jdbcx_dbcfg.java Com_stc_jdbcx_dbcfg.xsc EmptyETD.jar EmptyETD.xbs EmptyETD.xsc
\etd\oracle\	Com_stc_jdbcx_oraclecfg.java Com_stc_jdbcx_oraclecfg.xsc
\sql\ewOracleApps\common\	fn_request_status.sql SB_COPY_UTILS_PKG.sql
\sql\ewOracleApps\ewom\common\	sb_errors_pkg.sql sb_install.sql sb_initialize.sql
\sql\ewOracleApps\ewom\Inventory\ Customer_Item\	fn_request_customer_item.sql SB_CUST_UTILS_PKG.sql sb_validate_cust_items_pkg.sql
\sql\ewOracleApps\ewom\Inventory\ Customer_Item_Xref\	fn_request_cixref_import.sql SB_CUST_XREF_UTILS_PKG.sql SB_VALIDATE_CI_XREFS_PKG.sql
\sql\ewOracleApps\ewom\Inventory\ Cycle_Count\	fn_request_cyclecount_import.sql SB_CCENT_UTILS_PKG.sql sb_validate_cc_entries_pkg.sql
\sql\ewOracleApps\ewom\Inventory\ Item_Import\	fn_request_item_import.sql SB_ITM_UTILS_PKG.sql sb_validate_invrev_pkg.sql sb_validate_items_pkg.sql
\sql\ewOracleApps\ewom\Inventory\ Replenish\	fn_request_replenishment.sql SB_REPL_UTILS_PKG.sql sb_validate_rep_hdrs_pkg.sql sb_validate_rep_line_pkg.sql sb_validate_rep_pkg.sql
\sql\ewOracleApps\ewom\Inventory\ Transaction\	fn_request_Item_Transactions.sql SB_TRAN_UTILS_PKG.sql sb_validate_serial_pkg.sql sb_validate_tran_all_pkg.sql sb_validate_trans_lots_pkg.sql sb_validate_trans_pkg.sql
\sql\ewOracleApps\ewom\Purchase\ Order_Import\	fn_request_po_import.sql SB_POIMP_UTILS_PKG.sql sb_validate_poi_hdrs_pkg.sql sb_validate_poi_lines_pkg.sql sb_validate_poi_pkg.sql

Table 1 Participating Host & Registry Host

Subdirectories	Files
\\sql\ewOracleApps\ewom\Purchase\ Order_Receiving\	fn_request_po_receiving.sql SB_PO_RCV_UTILS_PKG.sql sb_validate_rcv_hdr_pkg.sql sb_validate_rcv_pkg.sql sb_validate_rcv_trans_pkg.sql
\\sql\ewOracleApps\ewom\Purchase\ Order_Requisition\	fn_request_po_requisitions.sql SB_PO_REQ_UTILS_PKG.sql sb_validate_por_req_pkg.SQL
\\ThirdParty\oracle\classes\	classes12.zip
\\ThirdParty\sun\classes\	jdbc2_0-stdext.jar

By default, the InstallShield installer also installs the following files within the \\eGate\Server\registry\repository\default tree on the Registry Host.

Table 2 Registry Host Only

Subdirectories	Files
\	addonlog.ctl connectionpoint.ctl convert.ctl corelog.ctl ewom.ctl stcewjdbcx.ctl stceworacle.ctl

2.5 Registering the DataSource

Before you can configure your Oracle e*Way, you need to test the connectivity.

To register your datasource

- 1 Install the Oracle ODBC drivers, following the installation instructions provided with the drivers.
- 2 On the task bar, click **Start** and then **Control Panel**.
- 3 Click **Administrative Tools**.
- 4 Click **Data Sources (ODBC)**.
- 5 In the *ODBC Data Source Administrator* window, click the **User DSN** tab.
- 6 Click **Add**.
- 7 Select **Oracle ODBC** and click **Finish**.
- 8 In the *OracleX ODBC Driver Setup* window (see Figure 10) enter:
 - A Data Source Name
 - B Description
 - C Service Name
 - D UserID
 - E Any additional system information for your Oracle installation

Figure 10 OracleX ODBC Driver Setup

- 9 Click **OK**.

2.6 ODBC Test

ODBC Test is a utility available from Microsoft. To use this utility:

- Download ODBC Test from <http://www.microsoft.com>
- Follow the online installation instructions provided by Microsoft

2.7 Optional Example Files

The installation CD contains two sample schemas, **POImport** and **POReceiving**, located in the `samples\ewom` directory. Also included are files representing valid and invalid data for testing purposes (see Table 3).

Table 3 samples\ewom Directory on CD-ROM

Subdirectories	Files
\POImport	POImportHeaders_Invalid.in POImportHeaders_Valid.in POImportLines_Invalid.in POImportLines_Valid.in POImportSample.zip readme.txt
\POReceiving	POReceivingSample.zip PORecHeaders_Invalid.in PORecHeaders_Valid.in PORecTrans_Invalid.in PORecTrans_Valid.in readme.txt

To use these schemas, you must load them onto your system using the procedure described on the following pages. See [Sample Schemas](#) on page 51 for descriptions of the sample schemas and instructions regarding their use.

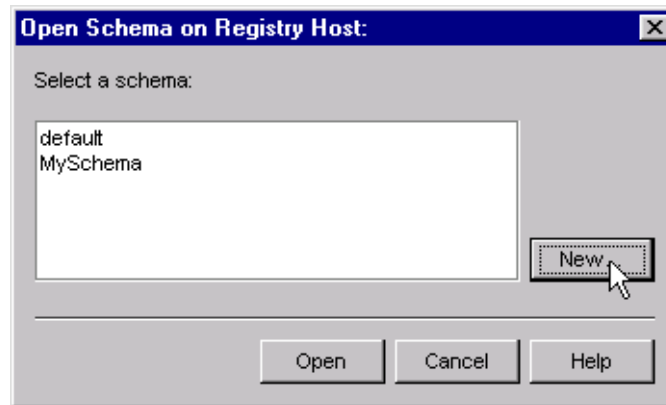
Note: *These schemas were created for testing purposes, and do not represent the optimal way of implementing this e*Way.*

2.7.1 Installation Procedure

To load a sample schema

- 1 Invoke the **Open Schema** dialog box and select **New** (see Figure 11).

Figure 11 Open Schema Dialog



- 2 Type the name you want to give to the schema (for example, **POImport.Sample**)
- 3 Select **Create from export** and navigate to the directory containing the sample schema by clicking the **Find** button.
- 4 Select the desired archive file (*.zip) and click **Open**.

Note: *The schema installs with the host name **localhost** and control broker name **localhost_cb**. If you want to assign your own names, copy the file *.zip to a local directory and extract the files. Using a text editor, edit the file *.exp, replacing all instances of the name **localhost** with your desired name. Add the edited .exp file back into the .zip file.*

2.7.2 Subdirectories and Files

The preceding procedure creates the following subdirectories and installs the following files within the `\eGate\Server\registry\repository\<SchemaName>` tree on the Registry Host, where `<SchemaName>` is the name you have assigned to the schema in step 2.

Table 4 Subdirectories and Files - POImport

Subdirectories	Files
\	POImport_imported.ctl
\runtime\collaboration_rules\POImport\	POImport.class POImport.ctl POImport.java POImport.xpr POImport.xts POImportBase.class
\runtime\configs\messageservice\	JMS_eWc.cfg JMS_eWc.sc
\runtime\configs\oracle\	Oracle11i_eWc.cfg Oracle11i_eWc.sc
\runtime\configs\ewfile\	HeadersErrorsOutput.cfg HeadersErrorsOutput.sc HeadersInput.cfg HeadersInput.sc LinesErrorsOutput.cfg LinesErrorsOutput.sc LinesInput.cfg LinesInput.sc
\sandbox\etd\POImport\	GenericOut.jar GenericOut.ssc GenericOut.xsc HeaderInput.jar HeaderInput.ssc HeaderInput.xsc LinesInput.jar LinesInput.ssc LinesInput.xsc newPOImportETD.jar newPOImportETD.xsc POImportETD.jar POImportETD.xsc

Table 5 Subdirectories and Files - POReceiving

Subdirectories	Files
\	POReceiving_imported.ctl
\runtime\configs\oracle\	kobe_eWc.cfg kobe_eWc.sc Oracle11i_eWc.cfg Oracle11i_eWc.sc
\runtime\configs\ewfile\	HeadersErrorsOutput.cfg HeadersErrorsOutput.sc HeadersInput.cfg HeadersInput.sc TransErrorsOutput.cfg TransErrorsOutput.sc TransInput.cfg TransInput.sc
\sandbox\collaboration_rules\POReceiving\	POReceiving.class POReceiving.ctl POReceiving.java POReceiving.xpr POReceiving.xts POReceivingBase.class
\sandbox\configs\oracle\	kobe_eWc.cfg kobe_eWc.sc
\sandbox\etd\POReceiving\	GenericOut.jar GenericOut.ssc GenericOut.xsc HeaderInput.jar HeaderInput.ssc HeaderInput.xsc PORecETD.jar PORecETD.xsc TransInput.jar TransInput.ssc TransInput.xsc

System Implementation

In this chapter we summarize the procedures required for implementing a working system incorporating the Java e*Way Intelligent Adapter for Oracle Manufacturing. Please see the *e*Gate Integrator User's Guide* for additional information.

3.1 Oracle Setup

Before operating the Oracle Manufacturing e*Way to import data into an Oracle application, the Oracle system must be set up properly. As an example, the procedure for implementing the Oracle Purchasing module is outlined below. Other Oracle modules have implementation procedures that are similar in nature, but different in detail. For detailed descriptions of the Oracle Applications Implementation Wizard (AIW) reference levels, see the *Oracle Applications Implementation Wizard User's Guide*.

Implementation Step	Req'd/Opt	AIW Reference
1. Set up System Administrator	Required	Common Applications
2. Define Accounting Key FlexFields	Required	Common Applications
3. Set up Calendars, Currencies, and Set of Books	Required	Common Applications
4. Define Human Resources Key FlexFields	Required	Common Applications
5. Define Locations	Required	Common Applications
6. Define Organizations and Organization Relationships	Required	Common Applications
7. Convert to a Multi-Org Architecture	Optional	Common Applications
8. Define Inventory Key FlexFields	Required	Common Applications
9. Define Units of Measure	Required	Common Applications
10. Define Freight Carriers	Optional	Common Applications
11. Define Item Attributes, Codes, and Templates	Required*	Common Applications
12. Define Categories	Required	Common Applications
13. Define Catalog Groups	Optional	Common Applications
14. Set up Personnel	Required	Common Applications
15. Set up Oracle Workflow	Required	Common Applications
16. Decide how to use the Account Generator	Required	Oracle Purchasing
17. Open Inventory and Purchasing Accounting Periods	Required	Common Distribution

Implementation Step	Req'd/Opt	AIW Reference
18. Define Subinventory Locations	Optional	Common Distribution
19. Define Cross-Reference Types	Optional	Oracle Purchasing
20. Define Tax Codes	Optional	Common Financial
21. Define Payment Terms	Optional	Common Financial
22. Set up Approval Information	Required	Oracle Purchasing
23. Define Lookups and Classes	Required*	Oracle Purchasing
24. Define Standard Attachments	Optional	Oracle Purchasing
25. Define Purchasing Options	Required	Oracle Purchasing
26. Define Buyers	Required	Oracle Purchasing
27. Define Items	Optional	Oracle Purchasing
28. Define Line Types	Required*	Oracle Purchasing
29. Start the Purchasing Database Administrator	Required	Oracle Purchasing
30. Define Financial Options	Required	Common Financial
31. Define Transaction Reasons	Optional	Oracle Purchasing
32. Define Receiving Options	Required	Oracle Purchasing
33. Set up Transaction Managers and Resubmission Intervals	Required	Oracle Purchasing
34. Define Suppliers	Required	Common Financial
35. Set up Workflow Options	Required*	Oracle Purchasing
36. Submit Workflow-Related Processes	Required	Oracle Purchasing
37. Define Descriptive FlexFields	Optional	Common Applications
38. Set up Automatic Sourcing	Optional	Oracle Purchasing
39. Perform Additional System Administrator Setup	Required	Common Applications
40. Define Manufacturing System and User Profiles	Required	Oracle Purchasing

* with defaults

3.2 Oracle Implementation Notes

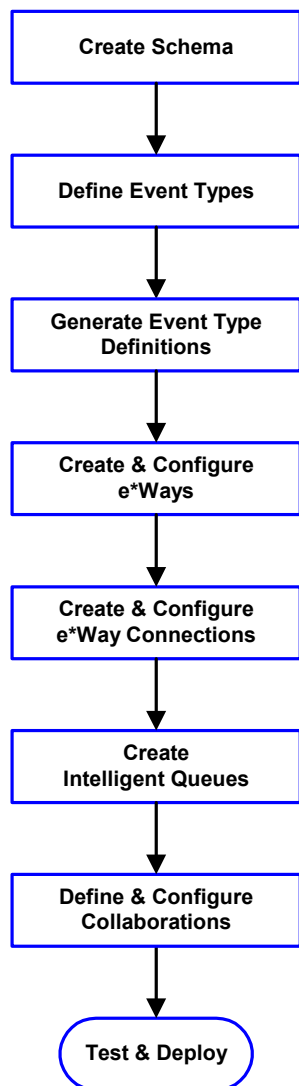
Included below is information regarding the Inventory module that is not currently covered in Oracle Manufacturing documentation.

- 1 The combination of **Customer_Id**, **Customer_Item_Number**, and **Address_Id** must be unique. You must guard against generating duplicates of this combination.
- 2 Any item that is validated through the Cycle Count Entry Interface must be cycle-count enabled in Inventory. The value of either **cycle_count_header_id** or **cycle_count_entry_id** must be populated. The value of **disable_date** must be null or greater than the sysdate.
- 3 It is mandatory that all items be imported to the parent organizational level before the importing to the child level; however, Oracle Inventory has the option of importing an item into all organizations while entering the data only for the parent organization.
- 4 Items already present in the base tables can be re-imported with changes by setting the process flag to **update**.

3.3 e*Gate Setup

This e*Way provides a specialized transport component for incorporation in an operational schema. The schema also contains Collaborations, linking different data or Event types, and Intelligent Queues. Typically, other e*Way types also are used as components of the schema.

3.3.1 Implementation Sequence



- 1 The first step is to create a new Schema—the subsequent steps apply only to this Schema (see [Creating a Schema](#) on page 39).
- 2 The second step is to define the Event Types you are transporting and processing within the Schema (see [Creating Event Types](#) on page 40).
- 3 Third, you need to associate the Event Types created in the previous step with Event Type Definitions (ETDs) derived from the applicable Business Rules (see [Generating Event Type Definitions](#) on page 41).
- 4 The fourth step is to create and configure the required e*Ways (see [Setting Up the e*Way](#) on page 69).
- 5 The fifth step is to configure the e*Way Connections (see [Schema Designer Creating e*Way Connections](#) on page 76).
- 6 Now you need to create Intelligent Queues to hold published Events (see [Creating Intelligent Queues](#) on page 47).
- 7 Next you need to define and configure the Collaborations between Event Types (see [Defining Collaborations](#) on page 48).
- 8 Finally, you must test your Schema. Once you have verified that it is working correctly, you may deploy it to your production environment.

3.3.2 Viewing e*Gate Components

Use the Navigator and Editor panes of the e*Gate Schema Designer to view the various e*Gate components. Note that you may only view components of a single schema at one time, and that all operations apply only to the current schema. All procedures in this chapter should be performed while displaying the **Components** Navigator pane. See the *e*Gate Integrator User's Guide* for a detailed description of the features and use of the Schema Designer.

3.4 Creating a Schema

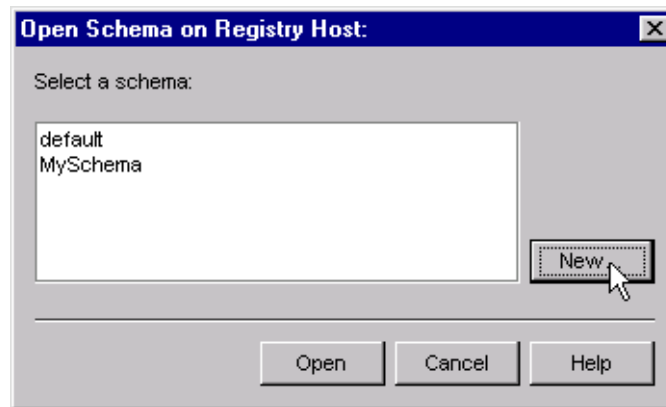
A schema is the structure that defines e*Gate system parameters and the relationships between components within the e*Gate system. Schemas can span multiple hosts.

Because all setup and configuration operations take place within an e*Gate schema, a new schema must be created, or an existing one must be started before using the system. Schemas store all their configuration parameters in the e*Gate Registry.

To select or create a schema

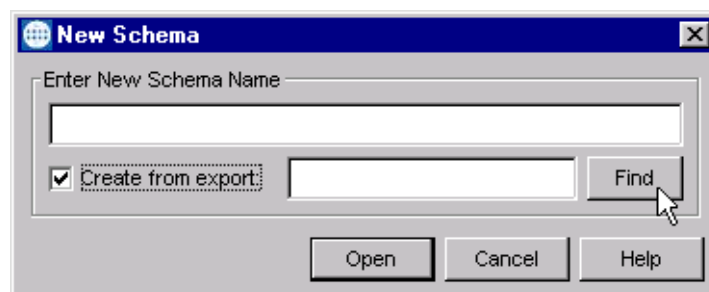
- 1 Invoke the **Open Schema** dialog box and **Open** an existing schema or click **New** to create a new schema.

Figure 12 Open Schema Dialog



- 2 Clicking **New** invokes the **New Schema** dialog box (Figure 13).

Figure 13 New Schema Dialog



- 3 Enter a new schema name and click **Open**.
- 4 The e*Gate Schema Designer then opens under your new schema name.
- 5 From the **Options** menu, click on **Default Editor** and select **Java**.
- 6 Select the **Components** tab, found at the bottom of the Navigator pane of the e*Gate Schema Designer window.
- 7 You are now ready to begin creating the necessary components for this new schema.

3.5 Creating Event Types

Within e*Gate, messages and/or packages of data are defined as Events. Each Event must be categorized into a specific Event Type within the schema.

To define the Event Types

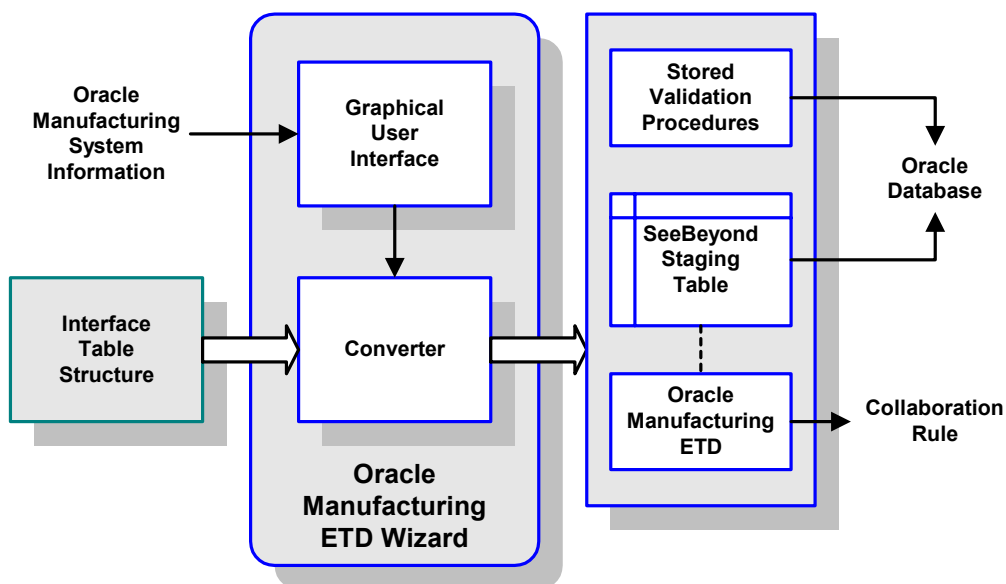
- 1 In the e*Gate Schema Designer's Navigator pane, select the **Event Types** folder.
- 2 On the Palette, click the **New Event Type** button.
- 3 In the **New Event Type Component** box, enter the name for the input Event Type and click **Apply**. Use this method to create all required Event Types, for example:
 - ♦ **InboundEvent**
 - ♦ **ValidEvent**
 - ♦ **InvalidEvent**
- 4 After you have created the final Event Type, click **OK**.

3.6 Generating Event Type Definitions

3.6.1 Using the ETD Builder

The Oracle Manufacturing e*Way incorporates a Wizard-based ETD Builder that helps customize the e*Way to your Oracle Manufacturing implementation. The Wizard obtains the hierarchies of the interface tables for the particular module, and creates an ETD that is used within the Collaboration Rule. It also sets up the corresponding staging table and the stored validation procedures to be run against the table. More details can be found in [Generating an ETD](#) on page 86.

Figure 14 Oracle Manufacturing ETD Builder



To run the Oracle Manufacturing ETD Builder

- 1 Click the New icon in the Java ETD Editor GUI. The New Event Type Definitions window appears, displaying all installed ETD Wizards
- 2 Invoke the Oracle Manufacturing ETD Wizard by clicking its icon (Figure 15). The Introduction dialog box appears (Figure 16).

Figure 15 Oracle Manufacturing Wizard Icon



Figure 16 Oracle Manufacturing Wizard Introduction



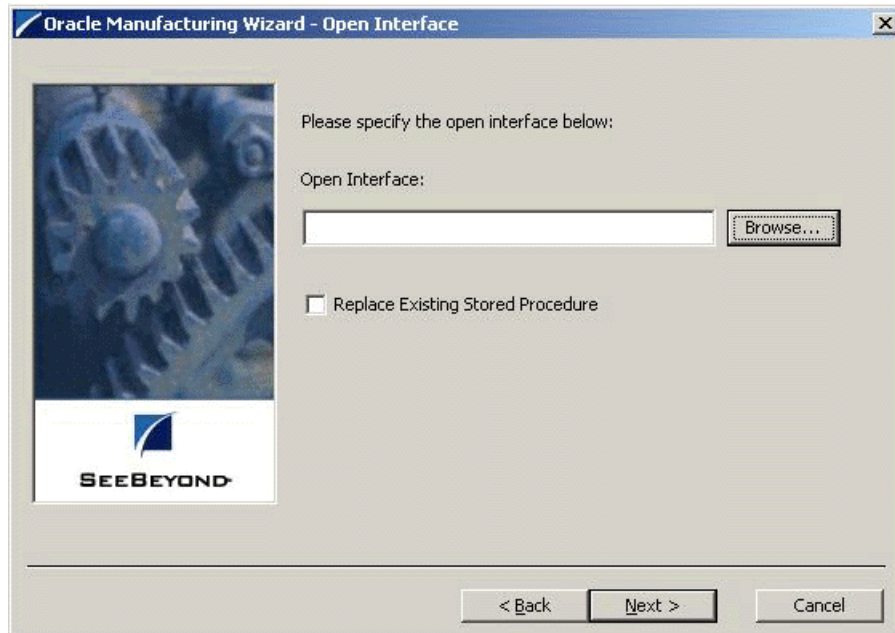
- 3 Click Next to continue to the Logon dialog box (Figure 17).

Figure 17 Oracle Manufacturing Wizard Logon Dialog



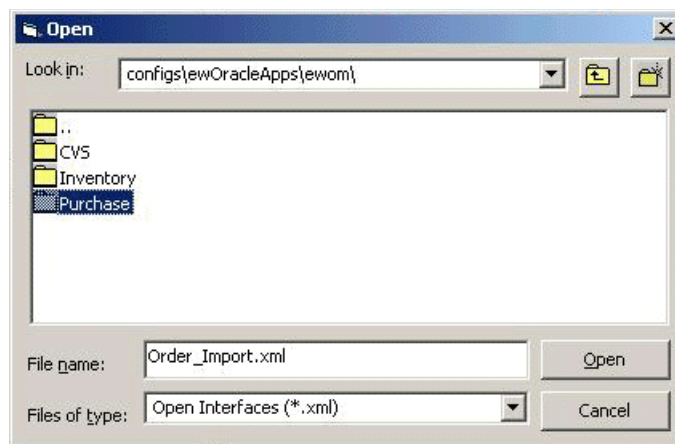
- 4 Select the data source from the drop-down list. In the second field enter your **User Name** and in the third field enter your **Password**.
- 5 Click **Next** to continue to the Open Interface dialog box (Figure 18).

Figure 18 Oracle Manufacturing Wizard Open Interface Dialog



- 6 Use the Browse facility to locate the XML file for the desired Open Interface. For example, select Order_Import (see Figure 19) and click **Open**.

Figure 19 Open File Dialog

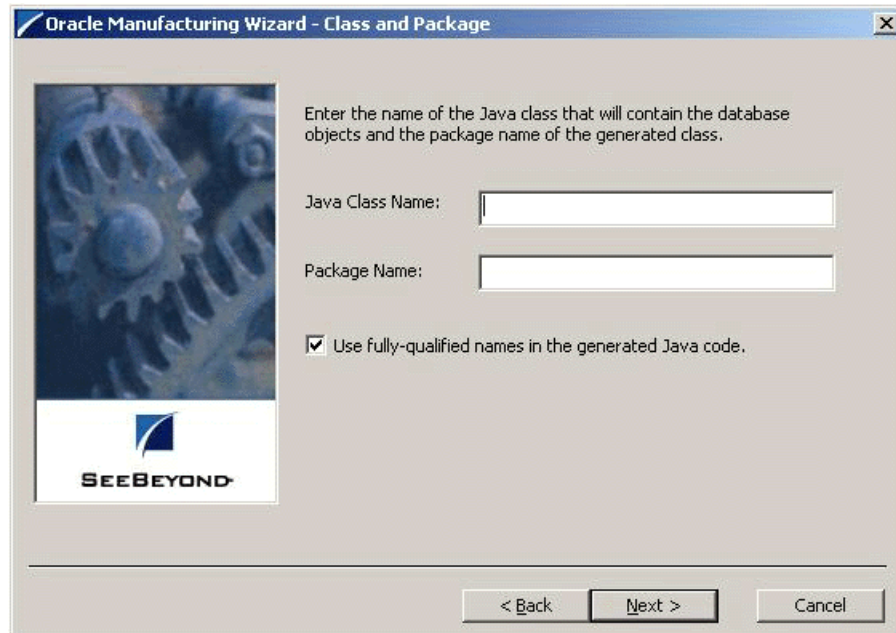


- 7 Usually, you will want to replace any existing stored procedures, so you should select the check box (in Figure 18). If you clear the check box, and any of the procedures already exist, the ETD will not be built and you will receive an error.

Note: Make sure you do not overwrite any procedures you have customized, without having retained backup copies.

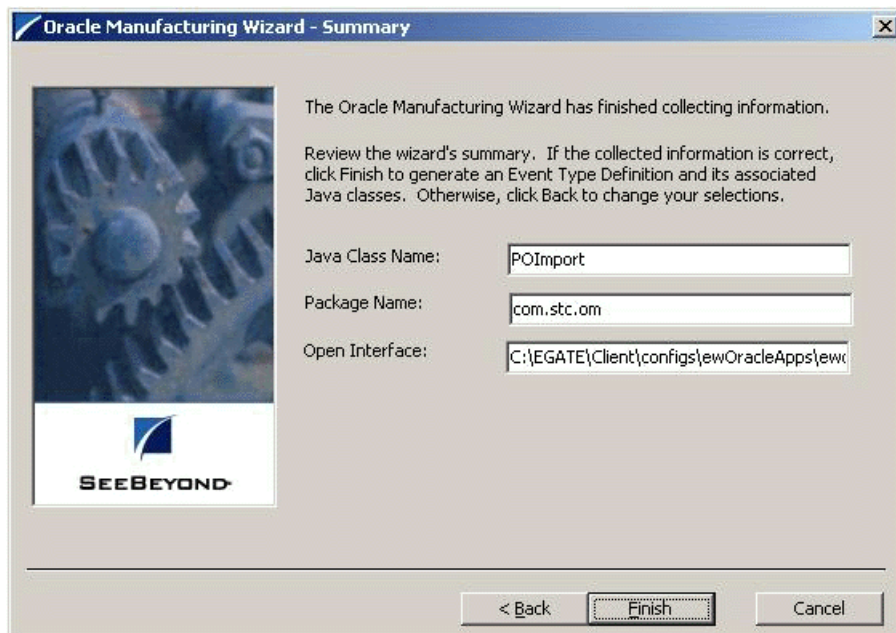
- 8 Click **Next** to continue to the Class and Package dialog box (Figure 20).

Figure 20 Oracle Manufacturing Wizard Class and Package Dialog



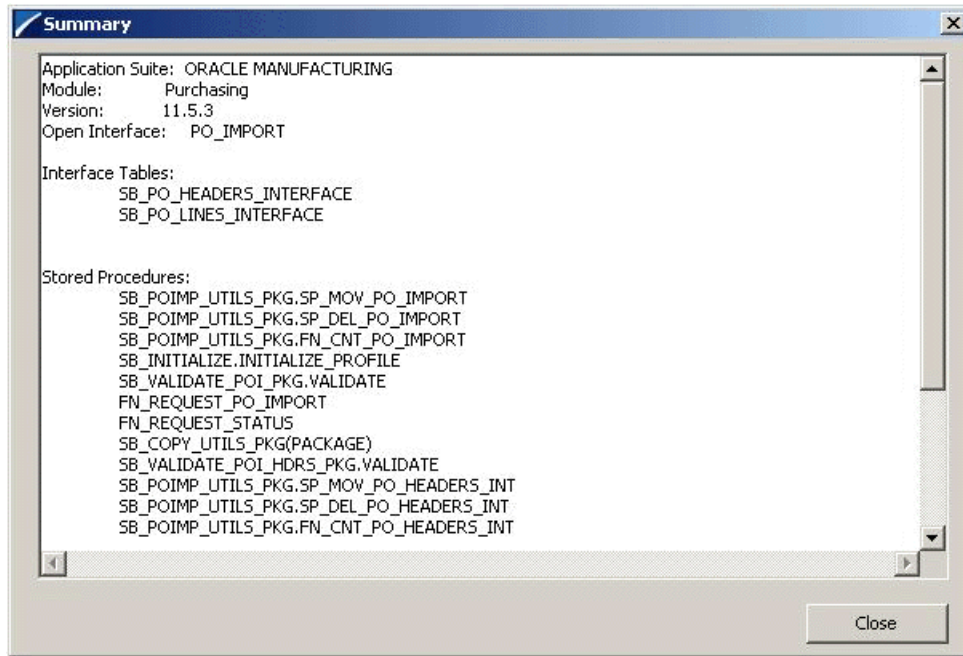
- 9 Type the name of the Java Class that will contain the database objects and the Package Name for the generated class into the respective text boxes.
- 10 Click Next to see a summary of your entries (Figure 21). If everything is correct, click Finish to generate the new ETD.

Figure 21 Oracle Manufacturing Wizard Summary



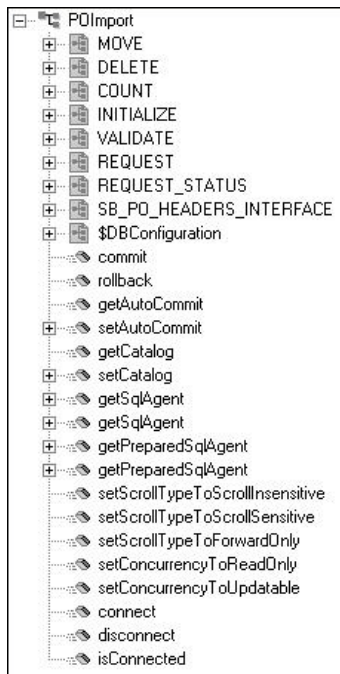
- 11 After the ETD is built, you are presented with a summary of its contents (Figure 22).

Figure 22 Summary Window



- 12 Click Close to return to the e*Gate Schema Designer, which now displays the ETD (see Figure 23). See [Exposed ETD Nodes](#) on page 88 for descriptions of the nodes.

Figure 23 ETD Structure Example



3.7 Assigning ETDs to Event Types

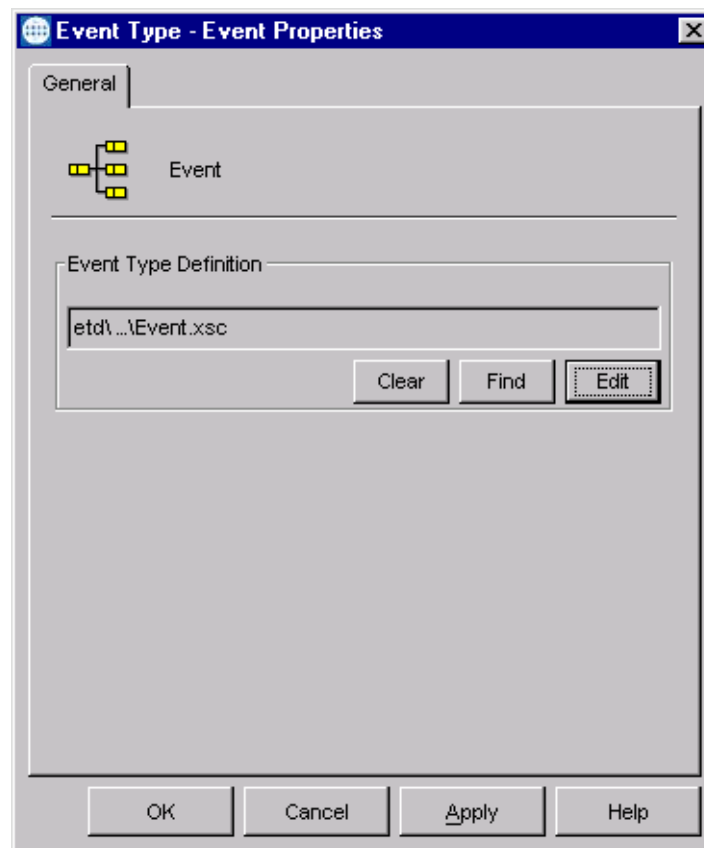
After you have created the e*Gate system's ETD files, you can assign them to Event Types you have already created.

To assign ETDs to Event Types

- 1 In the Schema Designer window, select the **Event Types** folder in the Navigator/Components pane.
- 2 In the Editor pane, select one of the Event Types you created.
- 3 Right-click on the Event Type and select **Properties**.

The Event Type Properties dialog box appears. See Figure 24.

Figure 24 Event Type Properties Dialog Box



- 4 Under Event Type Definition, click **Find**.

The Event Type Definition Selection dialog box appears; it is similar to the Windows Open dialog box.

- 5 Open the **etd** folder, then select the desired file name (.xsc). Note that there may be an intervening sub-folder.
- 6 Click **Select**. The file populates the Event Type Definition field.

- 7 To save any work in the properties dialog box, click **Apply** to enter it into the system.
- 8 When finished assigning ETDs to Event Types, click **OK** to close the properties dialog box and apply all the properties.

Each Event Type is associated with the specified Event Type Definition.

3.8 Creating Intelligent Queues

IQs are components that provide nonvolatile storage for Events within the e*Gate system as they pass from one component to another. IQs are *intelligent* in that they are more than just a “holding tank” for Events. They actively record information about the current state of Events.

Each schema must have an IQ Manager before you can add any IQs to it. You must create at least one IQ per schema for published Events within the e*Gate system. Note that e*Ways that publish Events externally do not need IQs.

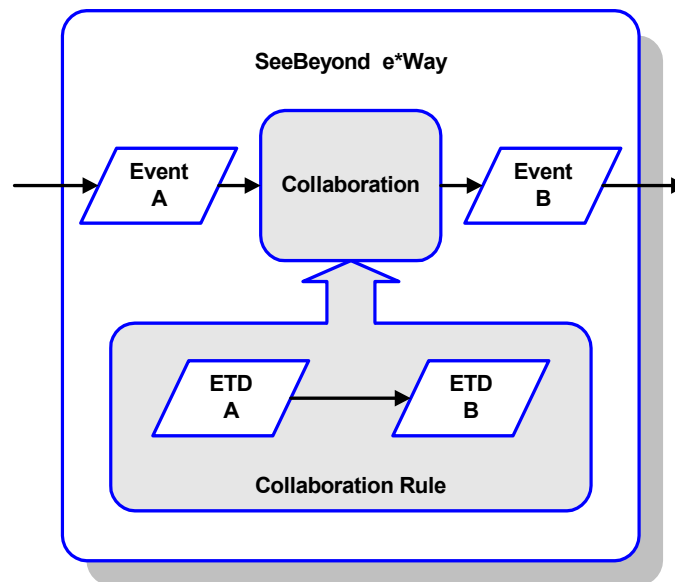
For more information on how to add and configure IQs and IQ Managers, see the *e*Gate Integrator System Administration and Operations Guide*. See the *e*Gate Integrator Intelligent Queue Services Reference Guide* and the *SeeBeyond JMS Intelligent Queue User’s Guide* for complete information on working with IQs.

3.9 Defining Collaborations

After you have created the required Event Type Definitions, you must define a Collaboration to transform the incoming Event into the desired outgoing Event.

Collaborations are e*Way components that receive and process Event Types, then forward the output to other e*Gate components. Collaborations consist of the Subscriber, which “listens” for Events of a known type or from a given source, and the Publisher, which distributes the transformed Event to a specified recipient. The same Collaboration cannot be assigned to more than one e*Gate component.

Figure 25 Collaborations



3.9.1 The Java Collaboration Rules Editor

Java Collaborations are defined using the e*Gate Java Collaboration Rules Editor. Note that the Java Collaboration environment supports multiple source and destination ETDs. The file extension for Java Collaboration Rules is **.xpr**. See the *e*Gate Integrator User's Guide* for descriptions of the Java Collaboration Rules Editor and its use.

3.9.2 Subcollaboration Rules

New methods are provided to allow you to use a Collaboration Rule as a parent or child of another Collaboration Rule. This allows you to insulate connectivity from transformation by separating out transformation-specific Collaboration Rules and maintaining them as individual units. See the *e*Gate Integrator User's Guide* for detailed information on Subcollaborations.

3.10 Running the e*Way

3.10.1 Configuring the Schema

Continue through the following procedures to configure the Oracle Manufacturing e*Way for your specific system. See [Schema Designer Creating e*Way Connections](#) on page 76 for more specific information on configuring the e*Way Connection parameters.

- 1 Open the desired schema in the e*Gate Schema Designer.
- 2 Open the e*Way Connection (**Oracle11i_eWc**) and edit the configuration file as follows:
 - ♦ Change the server name to your Oracle Manufacturing DB server name
 - ♦ Change the port number to your Oracle Manufacturing DB port number
 - ♦ Change the database name to your Oracle Manufacturing DB SID
 - ♦ Change the user name to your Oracle Manufacturing DB user name
 - ♦ Change the password to your Oracle Manufacturing DB password
- 3 From the File menu
 - A Click **Save** to save the new settings.
 - B Click **Promote to Run Time** to move the file to the Run-Time environment.
- 4 Click **OK** to acknowledge that the file has been promoted, and the Editor closes.
- 5 Click **OK** to close the e*Way Connection Properties dialog box.
- 6 You are now ready to run the sample schema.

3.10.2 Run the Schema

Running the sample Schema requires that a sample input file be created. Once the input file has been created, you can start the Control Broker from a command prompt to execute the Schema. After the Schema has been run, you can view the output text file to verify the results.

To start the Control Broker:

From a command prompt, type the following command (on one line):

```
stccb -ln logical_name -rh registry_host -rs schema_name -un  
      user_name -up password
```

where

- *logical_name* is the logical name of the Control Broker
- *registry_host* is the name of the Registry Host
- *schema_name* is the name of your schema

- *user_name* and *password* are a valid e*Gate username/password combination

3.11 Sample Schemas

Sample implementations, **POImport** and **POReceiving**, are available in the `\samples\ewom` directory of the e*Gate CD-ROM. These samples can be used to test your system following installation. See [Optional Example Files](#) on page 31 for installation instructions.

Note: *These schemas were created for testing purposes, and do not represent the optimal way of implementing this e*Way.*

These sample schemas are designed to wait for input for both staging tables, and then insert the records to the staging tables. Once data is received in both tables, the processing begins. Consequently, you should use only the invalid data or the valid data, not both at the same time. You can, however, combine the data into a single file—for instance, putting the valid and invalid Headers data into a single file called `HeadersData.HeadersIn`—to provide concurrent valid/invalid testing.

Note: *The Oracle Manufacturing e*Way must be properly installed on your system before you can run the sample schema.*

POImport

After importing the sample schema,

- 1 Configure the schema for your system (see [Configuring the Schema](#) on page 49).
- 2 Create a directory `C:\DATA\input\ewom` (you can change the location, if you also change the location specified in the schema).
- 3 To perform an *invalid data* test, copy the following files to the `\DATA\input\ewom` directory:
 - ♦ `POImportHeaders_Invalid.HeadersIn`
 - ♦ `POImportLines_Invalid.LinesIn`
- 4 Run the schema (see [Run the Schema](#) on page 49). These records will fail the pre-validation process, and send a detailed error message to the local directory specified in the outbound e*Way.
- 5 To perform a *valid data* test, copy the following files to the `\DATA\input\ewom` directory:
 - ♦ `POImportHeaders_Valid.HeadersIn`
 - ♦ `POImportLines_Valid.LinesIn`
- 6 Run the schema (see [Run the Schema](#) on page 49). These records will pass the pre-validation process, and be submitted by the Concurrent Manager.

PORceiving

After importing the sample schema,

- 1 Configure the schema for your system (see [Configuring the Schema](#) on page 49).
- 2 Create a directory `C:\DATA\input\ewom` (you can change the location, if you also change the location specified in the schema).
- 3 To perform an *invalid data* test, copy the following files to the `\DATA\input\ewom` directory:
 - ♦ `PORecHeaders_Invalid.HeadersIn`
 - ♦ `PORecTrans_Invalid.TransIn`
- 4 Run the schema (see [Run the Schema](#) on page 49). These records will fail the pre-validation process, and send a detailed error message to the local directory specified in the outbound e*Way.
- 5 To perform a *valid data* test, copy the following files to the `\DATA\input\ewom` directory:
 - ♦ `PORecHeaders_Valid.HeadersIn`
 - ♦ `PORecTrans_Valid.TransIn`
- 6 Run the schema (see [Run the Schema](#) on page 49). These records will pass the pre-validation process, and be submitted by the Concurrent Manager.

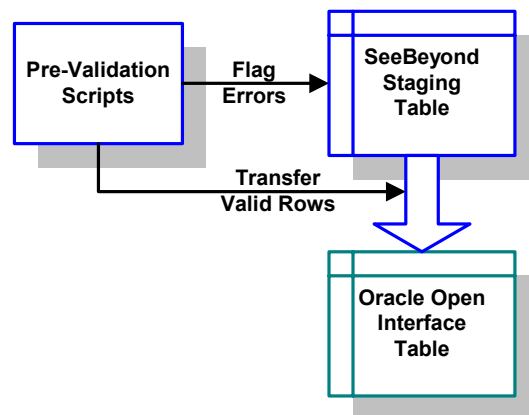
Pre-Validation Process

This chapter provides an overview of the pre-validation process used by the SeeBeyond Java e*Way Intelligent Adapter for Oracle Manufacturing. You can modify the pre-validation scripts as required to match your business rules. Brief descriptions of the individual scripts, including the error codes returned by the scripts, are found in [Validation Scripts](#) on page 165. Descriptions of the error code themselves are listed in [Error Codes](#) on page 213.

4.1 Overview

The various Oracle Inventory Management modules work together to provide a complete set of transactions and reports for maintaining inventory control. Each module is divided into sub-modules, each representing a logical division of workflow. Each sub-module uses at least one corresponding set of tables and scripts, as depicted in Figure 26.

Figure 26 Pre-Validation Subsystem



4.1.1 Pre-Validation Scripts

For Open Interfaces for which pre-validation scripts have been supplied, the SQL files shipped with the e*Way are sufficient for you to create a working schema. For Open Interfaces for which pre-validation scripts are not currently available, a template XML file (`templateXml.xml`) is supplied upon which you can build your own pre-validation package (see [Subdirectories and Files](#) on page 27). The procedure you need to follow is given in [Building a Custom Pre-Validation Package](#) on page 64.

Disclaimer

The pre-validation scripts packaged with the e*Way for the supported modules are rudimentary and do not cover all possible scenarios. Since the requirements from Oracle constantly change and evolve, you may be required to provide additional data and/or parameters to ensure that the Concurrent Manager request completes successfully, without errors.

4.1.2 Pre-Validation Procedures

Common Procedures

All pre-validation script packages (*_pkg.sql) contain the following procedures.

Table 6 Common Procedures

Procedure	Purpose
Initialize	Resets the staging table to default values.
TidyUp	Updates sb_pass_or_fail columns in the staging table from INTERMEDIATE to PASS for records that were not marked as failed by the pre-validation rules.
Validate	The main validation procedure, which calls the procedures that perform the pre-validations for the Interface Table.

Interface-Specific Procedures

A comprehensive list of all procedures and subroutines for currently-supported Open Interfaces can be found in [Validation Scripts](#) on page 165. These are grouped by script package (*_pkg), and reference the error codes generated by the specific subroutines. This information is also listed by error code in [Error Codes](#) on page 213.

4.2 Inventory Module

The Oracle Inventory module works with other Oracle Inventory Management modules to provide a complete set of transactions and reports for maintaining inventory control. This module is divided into the following sub-modules:

- Customer Items** on page 55
- Customer Items Cross-Reference** on page 56
- Cycle Count Entries** on page 57
- Items and Item Revisions** on page 58
- Replenishment** on page 59
- Transactions** on page 60

4.2.1 Customer Items

Oracle Interface Tables

Oracle provides the following Interface Tables for Customer Items:

- `mtl_ci_interface`

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- `sb_mtl_ci_interface`

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- `sb_validate_cust_items_pkg.sql`

4.2.2 Customer Items Cross-Reference

Oracle Interface Tables

Oracle provides the following Interface Tables for Customer Item Cross-References:

- `mtl_ci_xrefs_interface`

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- `sb_mtl_ci_xrefs_interface`

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- `sb_validate_ci_xrefs_pkg.sql`

4.2.3 Cycle Count Entries

Oracle Interface Table

Oracle provides the following Interface Table for Cycle Count Entries:

- `mtl_cc_entries_interface`

SeeBeyond Staging Table

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging table, corresponding to the Oracle Open Interface Table listed above:

- `sb_mtl_cc_entries_interface`

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Table listed above are packaged within the following files:

- `sb_validate_cc_entries_pkg.sql`

4.2.4 Items and Item Revisions

Oracle Interface Tables

Oracle provides the following Interface Tables for Items:

- mtl_system_items_interface
- mtl_item_revisions_interface

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- sb_mtl_system_items_interface
- sb_mtl_item_revisions_int

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- sb_validate_items_pkg.sql
- sb_validate_invrev_pkg.sql

4.2.5 Replenishment

Oracle Interface Tables

Oracle provides the following Interface Tables for Item Replenishment:

- mtl_replenish_headers_int
- mtl_replenish_lines_int

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- sb_mtl_replenish_headers_int
- sb_mtl_replenish_lines_int

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- sb_validate_rep_headers_pkg.sql
- sb_validate_rep_lines_pkg.sql

These files are controlled by the following master validation file:

- sb_validate_rep_pkg.sql

4.2.6 Transactions

Oracle Interface Tables

Oracle provides the following Interface Tables for Transactions:

- mtl_transactions_interface
- mtl_transaction_lots_interface
- mtl_serial_numbers_interface

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- sb_mtl_transactions_interface
- sb_mtl_transaction_lots_inter
- sb_mtl_serial_numbers_interfac

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- sb_validate_trans_pkg.sql
- sb_validate_trans_lots_pkg.sql
- sb_validate_serial_nos_pkg.sql

These files are controlled by the following master validation file:

- sb_validate_tran_all_pkg.sql

4.3 Purchase Order Import Module

The Oracle Purchase Order Import module works with other Oracle Inventory Management modules to provide a complete set of transactions and reports for maintaining inventory control. This module is divided into the following sub-modules:

Purchase Order Import on page 61

Purchase Order Receiving on page 62

Purchase Order Requisitions on page 63

Each sub-module uses at least one corresponding set of tables and scripts, as depicted in **Figure 26 on page 53**.

4.3.1 Purchase Order Import

Oracle Interface Tables

Oracle provides the following Interface Tables for Purchase Order Import:

- po_headers_interface
- po_lines_interface

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- sb_po_headers_interface
- sb_po_lines_interface

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- sb_validate_poi_hdrs_pkg.sql
- sb_validate_poi_lines_pkg.sql

These files are controlled by the following master validation file:

- sb_validate_poi_pkg.sql

4.3.2 Purchase Order Receiving

Oracle Interface Tables

Oracle provides the following Interface Tables for Receiving:

- rcv_headers_interface
- rcv_transactions_interface

SeeBeyond Staging Tables

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging tables, corresponding to the Oracle Open Interface Tables listed above:

- sb_rcv_headers_interface
- sb_rcv_transactions_interface

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Tables listed above are packaged within the following files:

- sb_validate_rcv_hdr_pkg.sql
- sb_validate_rcv_trans_pkg.sql

These files are controlled by the following master validation file:

- sb_validate_rcv_pkg.sql

4.3.3 Purchase Order Requisitions

Oracle Interface Table

Oracle provides the following Interface Table for Requisitions:

- `po_requisitions_interface_all`

SeeBeyond Staging Table

The e*Way Intelligent Adapter for Oracle Manufacturing sets up the following staging table, corresponding to the Oracle Open Interface Table listed above:

- `sb_po_requisitions_interface_a`

Pre-Validation Scripts

The scripts corresponding to the SeeBeyond Staging Table listed above are packaged within the following file:

- `sb_validate_por_req_pkg.sql`

4.4 Building a Custom Pre-Validation Package

To build a custom pre-validation package for a non-supported Open Interface

- 1 Copy the template XML file to the desired directory on the server. The file can be found in the following location:


```
eGate\server\registry\repository\configs\ewOracleApps\
  template\templateXml.xml
```
- 2 Replace the entry `<%% XXXX %%>` with the name specific to the Open Interface. The name should include the path relative to `eGate\client`.
- 3 The `initialize` and `request_status` scripts are already specified in the XML file. If you do not need them, simply remove them from the XML file. If you need to customize these scripts,
 - A Copy the SQL file to the desired directory.
 - B Rename the file—the package name must be the same as the SQL file name.
 - C Modify the scripts as needed for your own implementation.
 - D In the XML file, refer to these new customized SQL files instead of the shipped common SQL script.

Note: You should leave the `Initialize_Profile` stored procedure name as-is, and as the entry point for initialize package.

- 4 Run the ETD builder and specify the newly-created XML file.
- 5 The builder will automatically generate the UTILITY package, Concurrent Manager Request Function, and Validation Package. These SQL Scripts are generated in the directory specified in the XML (see step 2, above).
- 6 Since the newly-generated Validation package and Concurrent Manager Request Function are empty, you must now customize them according to your needs. See [Concurrent Manager Request Function](#), below.
- 7 Compile the customized SQL scripts in the Oracle database.

4.4.1 Concurrent Manager Request Function

The default SQL script template for Concurrent Manager Request has 100 identical input parameters for `FND_REQUEST.SUBMIT_REQUEST`. If you need to customize the input parameters for your generated script, the following tags can be used to specify the input parameters.

```
<Concurrent_Manager>sql\ewOracleApps\ewom\*\*.sql
</Concurrent_Manager>
<Request_Param>parameter1</Request_Param>
<Request_Param>parameter2</Request_Param>
<Request_Param>parameter3</Request_Param>
```


4.4.2 Template DTD

The Data Type Definition (DTD) associated with the template XML file is shown below:

```
<!ELEMENT OPEN_INTERFACE (Initialize_Script?, Validation_Script?,
Utility_Script, Concurrent_Manager, Request_Param*,
Concurrent_Manager_Status?, Interface_Table+, Error_Handle?,
Pre_Required_Script*)>

<!-- Open interface name -->
<!ATTLIST OPEN_INTERFACE Name CDATA #REQUIRED>

<!-- Oracle Manufacturing Version -->
<!ATTLIST OPEN_INTERFACE Version CDATA #REQUIRED>

<!-- Application Suite -->
<!ATTLIST OPEN_INTERFACE Application-Suite CDATA #REQUIRED>

<!-- Module Name -->
<!ATTLIST OPEN_INTERFACE Module CDATA #REQUIRED>

<!-- utiltiy store procedure name at root level, this attribute is
optional
if this attribute is specified, it will override the default
name convention, which is derived fromName attribute of
OPEN_INTERFACE
-->
<!ATTLIST OPEN_INTERFACE Util_Name CDATA #IMPLIED>

<!-- relative path to the sql script file for initialize package-->
<!ELEMENT Initialize_Script (#PCDATA)>

<!-- relative path to the sql script file for validation package -->
<!ELEMENT Validation_Script (#PCDATA)>

<!-- relative path to the sql script file for Utility package -->
<!ELEMENT Utility_Script (#PCDATA)>

<!-- relative path to the sql script file for concurrent manager
request function -->
<!ELEMENT Concurrent_Manager (#PCDATA)>

<!-- parmater description for concurrent manager request function --
>
<!ELEMENT Request_Param (#PCDATA)>

<!-- relative path to the sql script file for function to retrieve
concurrent manager request status -->
<!ELEMENT Concurrent_Manager_Status (#PCDATA)>

<!-- Oracle Open Interface table definition -->
<!ELEMENT Interface_Table (Validation_Script?, Interface_Table*)>

<!-- name for the oracle open interface table -->
<!ATTLIST Interface_Table Name CDATA #REQUIRED>

<!-- name for SB staging table, this attribute is optional
if this attribute is set, program will directly use
its value as the SB staging table for this specific
interface
-->
<!ATTLIST Interface_Table SB_Name CDATA #IMPLIED>
```

```
<!-- utiltiy store procedure name at specific interface level, this
attribute is optional
    if this attribute is specified, it will override the default name
convention, which is
    derived from the specific interface name
-->
<!ATTLIST Interface_Table Util_Name CDATA #IMPLIED>

<!-- Error handling definition -->
<!ELEMENT Error_Handle (Error_Table?, Error_PKG?)>

<!-- relative path to the sql script for creating error table -->
<!ELEMENT Error_Table (#PCDATA)>

<!-- relative path to the sql script for error handling -->
<!ELEMENT Error_PKG (#PCDATA)>

<!-- relative path to the sql script that need to be compiler in order
for other
    stored procedure to be compiled successfully -->
<!ELEMENT Pre_Required_Script (#PCDATA)>
```

4.5 Naming Conventions

Oracle Interface Name

The specific name for the Open Interface, which is specified in the XML file.

Short Table Name

The name used for the generated utility stored procedures, for each interface table. It is derived from the corresponding Oracle Interface Table name as follows:

If the Oracle Interface Table name is longer than 23 characters, replace INTERFACE or IFACE_ALL with INT to form the short table name.

(Since there is a limitation of 30 characters placed on the function/stored procedure name in Oracle, the Open Interface name specified in the XML file must be no more than 23 characters.)

Stored Procedures

The utility stored procedure names are derived from the Short Table name, according to the following convention:

Sb_XXX_<SHORT TABLE NAME>

See the descriptions given in [Exposed ETD Nodes](#) on page 88 for the exact names.

Staging Table Node

The name of the staging table node is derived from corresponding Oracle Interface Table name by adding the prefix **SB_** and truncating to 30 characters, if necessary.

SQL File Name

If the SQL file is used to create a packaged stored procedure, the file name must be the same as the package name.

Concurrent Manager Request Function

The function name is derived from the Open Interface name specified in the XML, and has the form: FN_REQUEST_<ORACLE INTERFACE NAME>.

e*Way Setup

This chapter describes the procedures required to customize the SeeBeyond Java e*Way Intelligent Adapter for Oracle Manufacturing to operate within your production system.

5.1 Overview

After creating a schema, you must instantiate and configure the e*Way Intelligent Adapter for Oracle Manufacturing to operate within the schema. A wide range of setup options allow the e*Way to conform to your system's operational characteristics and your facility's operating procedures.

The topics discussed in this chapter are:

[Setting Up the e*Way](#) on page 69

[Schema Designer Creating e*Way Connections](#) on page 76

[Managing e*Way Connections](#) on page 79

[Using the e*Way Configuration Editor](#) on page 81

[Troubleshooting the e*Way](#) on page 84

5.2 Setting Up the e*Way

Note: The e*Gate Schema Designer GUI runs only on the Windows operating system.

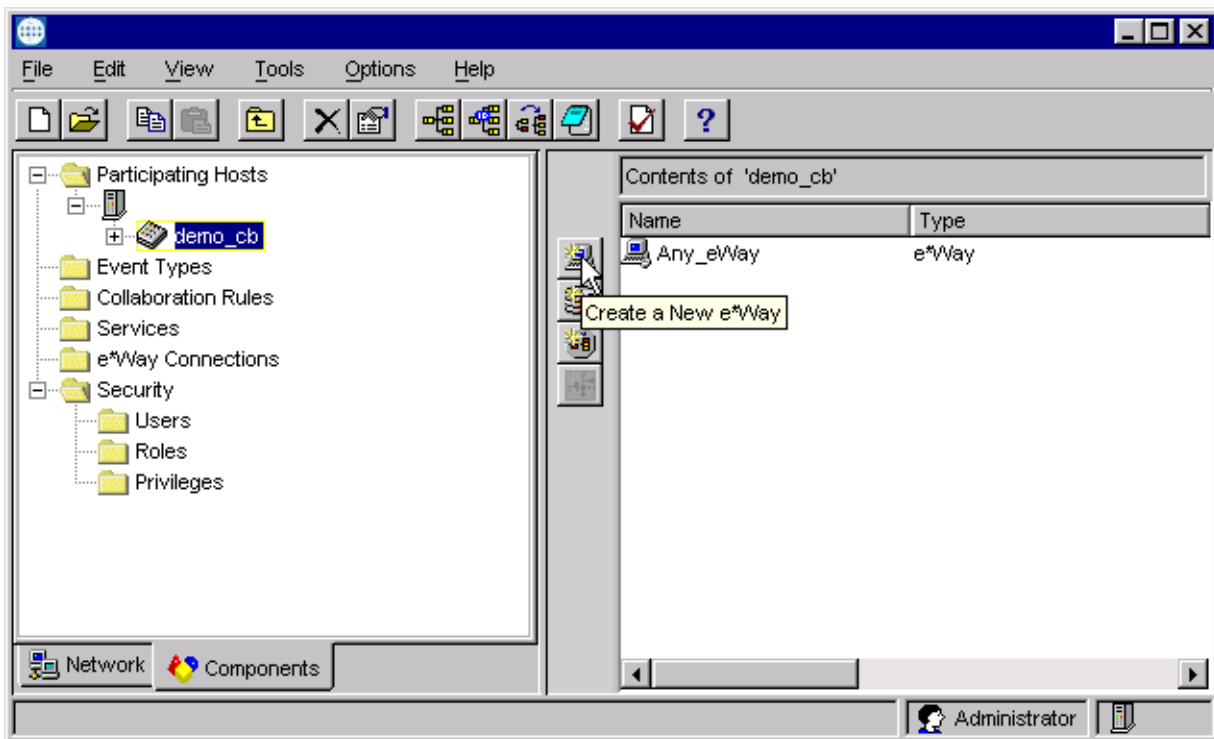
5.2.1 Creating the e*Way

The first step in implementing an e*Way is to define the e*Way component using the e*Gate Schema Designer.

To create an e*Way

- 1 Open the schema in which the e*Way is to operate.
- 2 Select the e*Gate Schema Designer Navigator's **Components** tab.
- 3 Open the host on which you want to create the e*Way.
- 4 Select the Control Broker you want to manage the new e*Way.

Figure 27 e*Gate Schema Designer Window (Components View)



- 5 On the Palette, click **Create a New e*Way**.
- 6 Enter the name of the new e*Way, then click **OK**.
- 7 All further actions are performed in the e*Gate Schema Designer Navigator's **Components** tab.

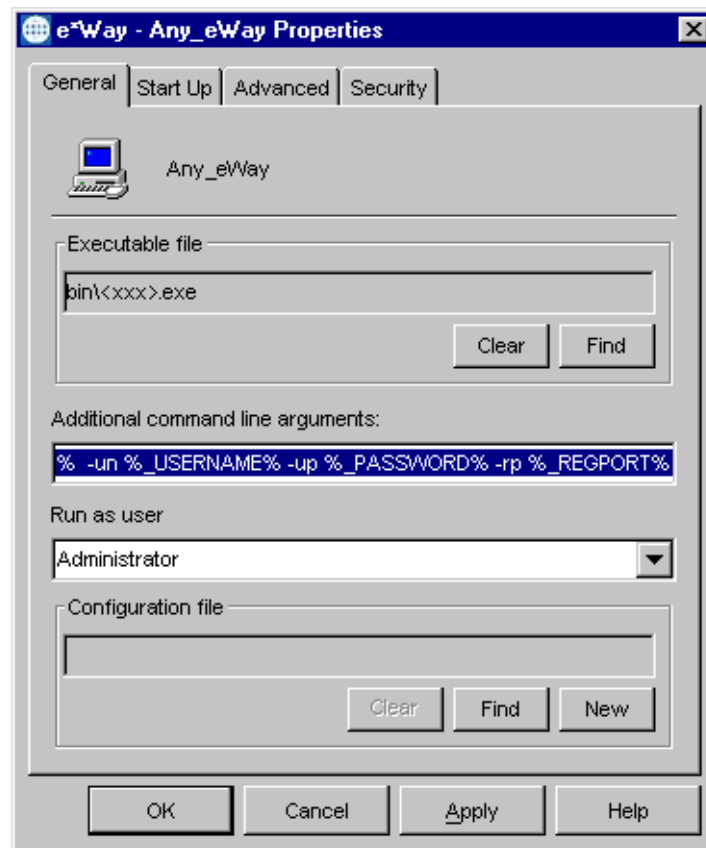
5.2.2 Modifying e*Way Properties

To modify any e*Way properties

- 1 Right-click on the desired e*Way and select **Properties** to edit the e*Way's properties. The properties dialog opens to the **General** tab (shown in Figure 28).

Note: The executable and default configuration files used by this e*Way are listed in **e*Way Components** on page 18.

Figure 28 e*Way Properties (General Tab)



- 2 Make the desired modifications, then click **OK**.

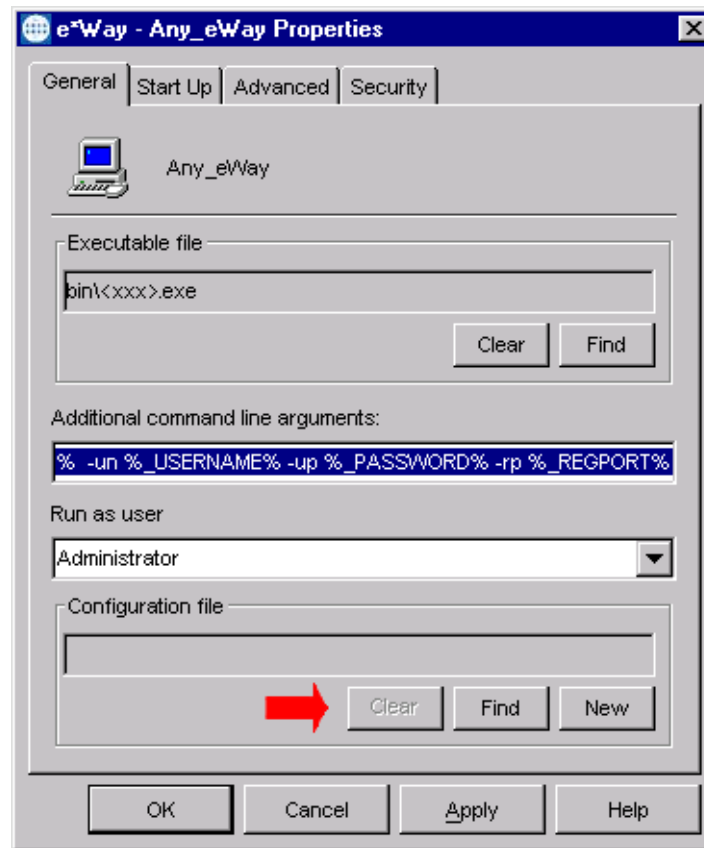
5.2.3 Configuring the e*Way

The e*Way's inherent configuration parameters are stored in an ASCII text file with a .def extension. The e*Way Editor provides a simple graphical interface for viewing and changing those parameters to create a working configuration (.cfg) file.

To change e*Way configuration parameters

- 1 In the e*Gate Schema Designer's Component editor, select the e*Way you want to configure and display its properties.

Figure 29 e*Way Properties - General Tab



- 2 Under **Executable File**, click **Find** to locate `stceway.exe`.
- 3 Under **Configuration File**, click **New** to create a new file or **Find** to select an existing configuration file. If you select an existing file, an **Edit** button appears; click the button to edit the currently selected file.
- 4 You now are in the e*Way Configuration Editor (see [Using the e*Way Configuration Editor](#) on page 81). The e*Way's configuration parameters are described in [Chapter 7](#).

5.2.4 Changing the User Name

Like all e*Gate executable components, e*Ways run under an e*Gate user name. By default, all e*Ways run under the **Administrator** user name. You can change this if your site's security procedures so require.

To change the user name

- 1 Display the e*Way's properties dialog.
- 2 On the **General** tab, use the **Run as user** list to select the e*Gate user under whose name you want this component to run.

See the *e*Gate Integrator System Administration and Operations Guide* for more information on the e*Gate security system.

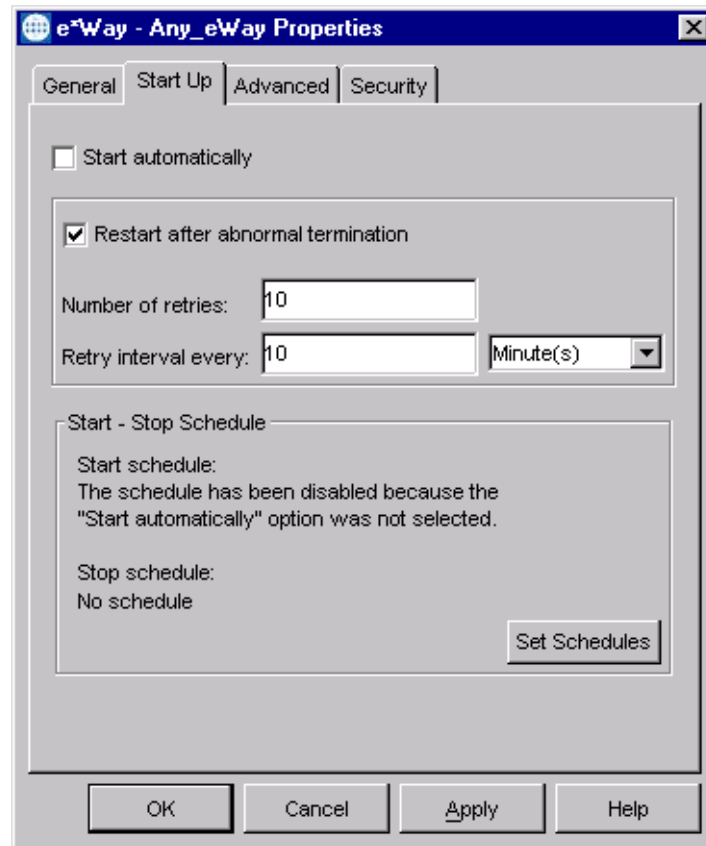
5.2.5 Setting Startup Options or Schedules

SeeBeyond e*Ways can be started or stopped by any of the following methods:

- The Control Broker can start the e*Way automatically whenever the Control Broker starts.
- The Control Broker can start the e*Way automatically whenever it detects that the e*Way terminated execution abnormally.
- The Control Broker can start or stop the e*Way on a schedule that you specify.
- Users can start or stop the e*Way manually using an interactive monitor.

You determine how the Control Broker starts or shuts down an e*Way using options on the e*Way properties **Start Up** tab (see Figure 30). See the *e*Gate Integrator System Administration and Operations Guide* for more information about how interactive monitors can start or shut down components.

Figure 30 e*Way Properties (Start-Up Tab)



To set the e*Way's startup properties

- 1 Display the e*Way's properties dialog.
- 2 Select the **Start Up** tab.
- 3 To have the e*Way start automatically when the Control Broker starts, select the **Start automatically** check box.
- 4 To have the e*Way start manually, clear the **Start automatically** check box.
- 5 To have the e*Way restart automatically after an abnormal termination:
 - A Select **Restart after abnormal termination**.
 - B Set the desired number of retries and retry interval.
- 6 To prevent the e*Way from restarting automatically after an abnormal termination, clear the **Restart after abnormal termination** check box.
- 7 Click **OK**.

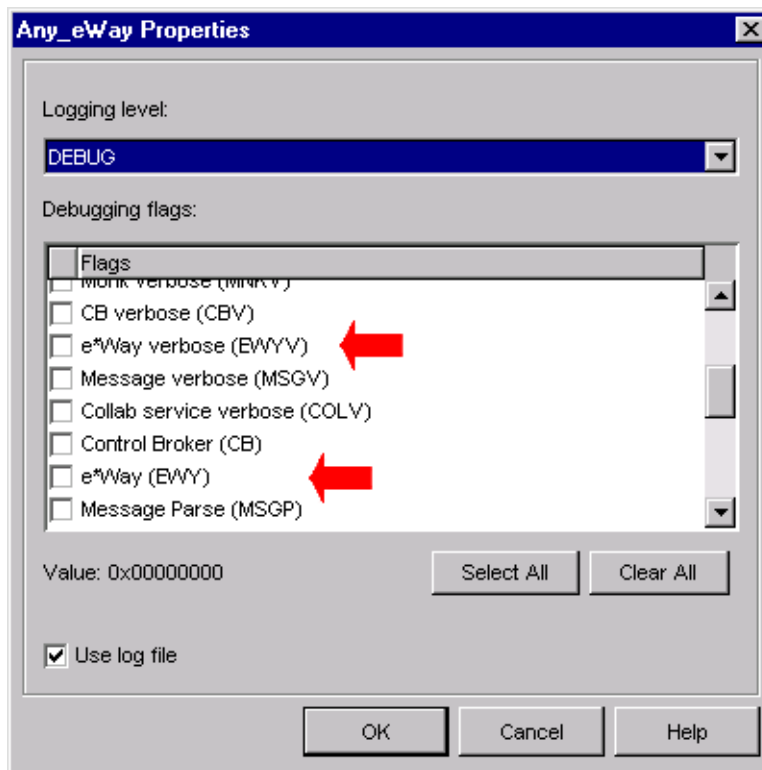
5.2.6 Activating or Modifying Logging Options

Logging options enable you to troubleshoot problems with the e*Way and other e*Gate components.

To set the e*Way debug level and flag

- 1 Display the e*Way's Properties dialog.
- 2 Select the **Advanced** tab.
- 3 Click **Log**. The dialog window appears, as shown in Figure 31.

Figure 31 e*Way Properties (Advanced Tab - Log Option)



- 4 Select **DEBUG** for the **Logging level**.
- 5 Select either **e*Way (EWY)** or **e*Way Verbose (EWYV)** for the **Debugging flag**. Note that the latter has a significant impact on system performance.
- 6 Click **OK**.

The other options apply to other e*Gate components and are activated in the same manner. See the *e*Gate Integrator Alert and Log File Reference* for additional information concerning log files, logging options, logging levels, and debug flags.

5.2.7 Activating or Modifying Monitoring Thresholds

Monitoring thresholds enable you to monitor the throughput of the e*Way. When the monitoring thresholds are exceeded, the e*Way sends a Monitoring Event to the Control Broker, which routes it to the Schema Manager and any other configured destinations.

- 1 Display the e*Way's properties dialog.
- 2 Select the **Advanced** tab.
- 3 Click **Thresholds**.
- 4 Select the desired threshold options and click **OK**.

See the *e*Gate Integrator Alert and Log File Reference* for more information concerning threshold monitoring, routing specific notifications to specific recipients, or for general information about e*Gate's monitoring and notification system.

5.3 Schema Designer Creating e*Way Connections

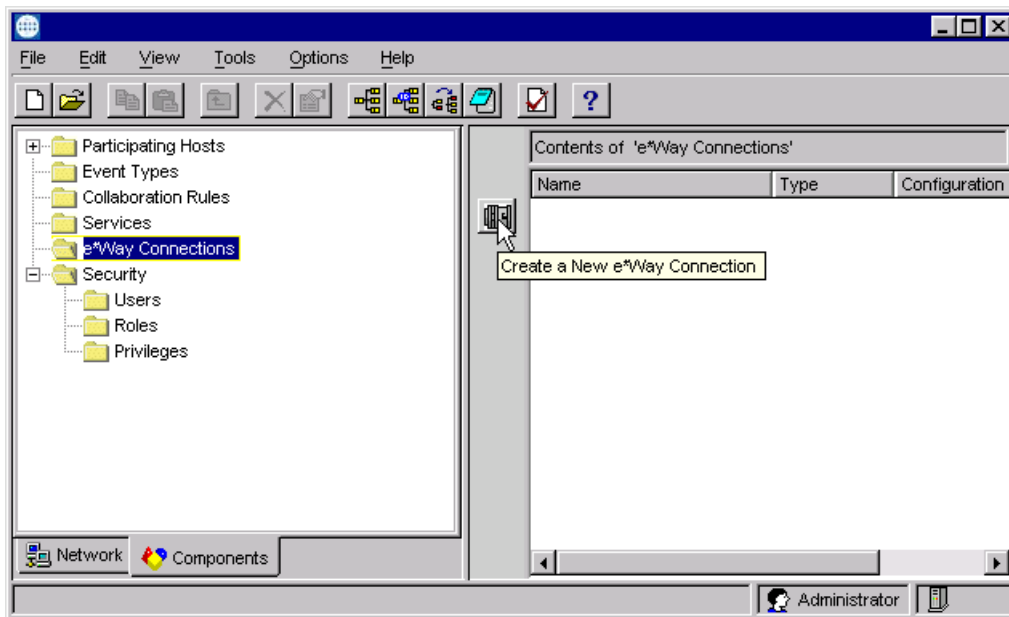
The e*Way Connections are created and configured in the Schema Designer.

Note: The e*Gate Schema Designer GUI runs only on the Windows operating system.

To create and configure the e*Way Connections

- 1 In the Schema Designer's Component editor, select the **e*Way Connections** folder.

Figure 32 Schema Designer - e*Way Connections Folder (1)




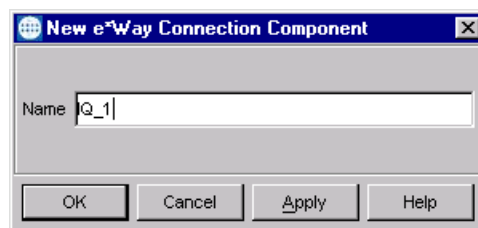
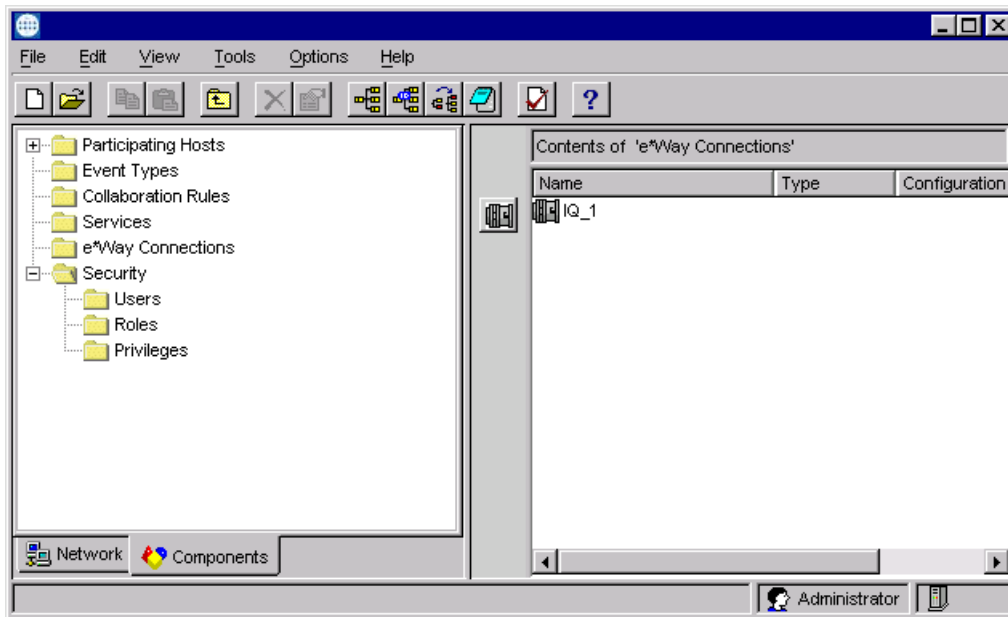
- 2 On the Palette, click the **Create a New e*Way Connection** button , which opens the New e*Way Connection Component dialog box.

Figure 33 New e*Way Connection Component Dialog Box



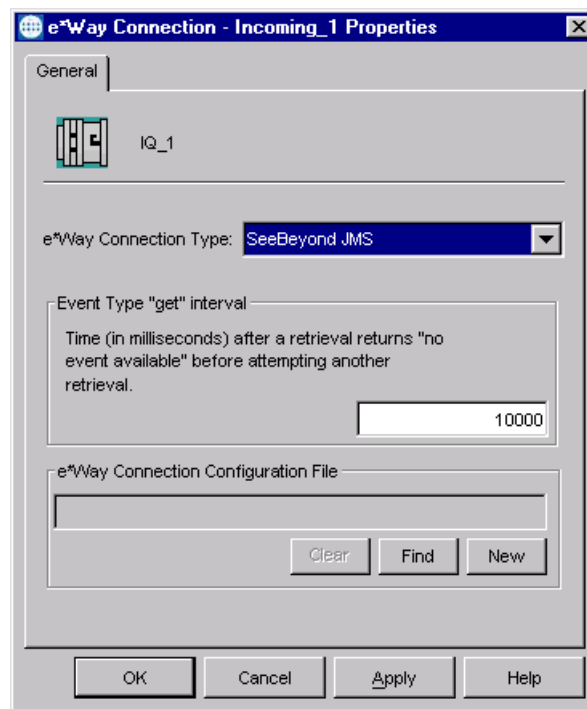
- 3 Enter a name for the e*Way Connection and click **OK**. The new e*Way Connection appears in the Schema Designer Contents pane.

Figure 34 Schema DesignerSchema Designer - e*Way Connections Folder (2)



- 4 Right-click the new e*Way Connection icon and select **Properties** to open the e*Way Connection Properties dialog box.

Figure 35 e*Way Connection Properties Dialog Box



- 5 From the e*Way Connection Type drop-down box, select **SeeBeyondJMS**.
- 6 Enter the Event Type *get* interval in the dialog box provided (optional).

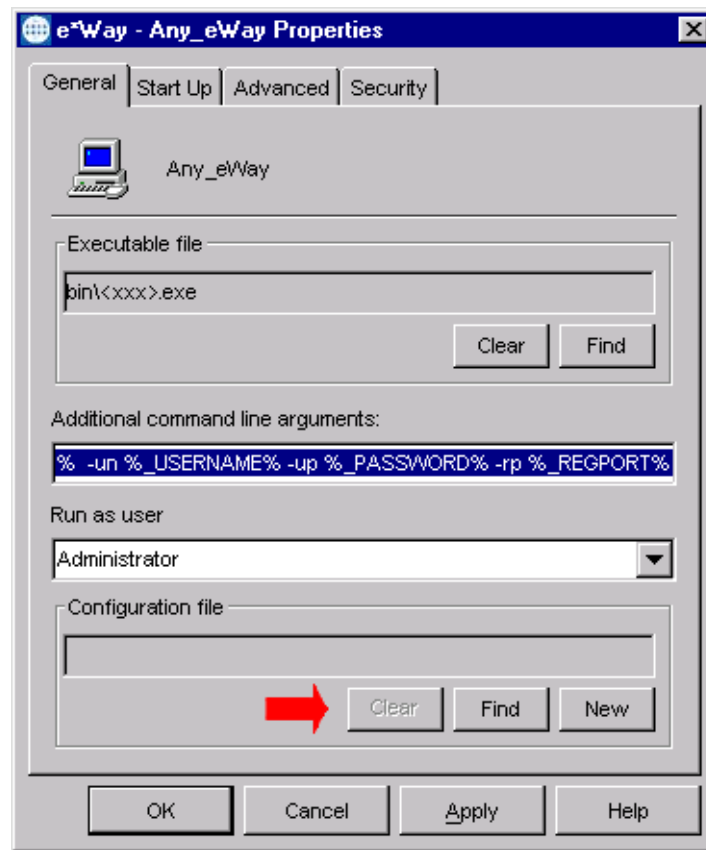
- 7 Click **New** to start the e*Way Connection Configuration File Editor, where you can create a new e*Way Connection Configuration File.

To modify the e*Way Connections

- 1 In the e*Gate Schema Designer’s Component editor, select the e*Way you want to reconfigure and display its properties.

Note: The executable and default configuration files used by this e*Way are listed in **e*Way Components** on page 18.

Figure 36 e*Way Properties - General Tab



- 2 Under **Configuration File**, click **New** to create a new file or **Find** to select an existing configuration file. If you select an existing file, an **Edit** button appears; click the button to edit the currently selected file.
- 3 You are now in the e*Way Configuration Editor (see **Using the e*Way Configuration Editor** on page 81). The e*Way Connection’s configuration parameters are described in **Chapter 7**.

Note: You must restart the e*Way after modifying the e*Way connection.

5.4 Managing e*Way Connections

The Connection Manager allows you to define the functionality of your e*Way connection. You can:

- Select when to initiate an e*Way connection
- Select when to close the e*Way connection and disconnect
- Select the status of your e*Way connection
- Have the Collaboration call the **OnConnectionDown** method when the connection fails

The Connection Manager is specifically designed to take full advantage of the enhanced e*Gate functionality. In e*Gate 4.5.1 or earlier, this enhanced functionality is visible, but ignored.

The Connection Manager is controlled as described in [Connector Settings](#) on page 106. If you choose to control the e*Way Connections manually, you may find the following chart helpful.

Table 7 e*Way Connection Controls

Method	Automatic	On-Demand	Manual
onConnectionUp	yes	no	no
onConnectionDown	yes	yes—but only if the connection attempt fails	no
Automatic Transaction (XA)	yes	no	no
Manual Transaction (XA)	yes	no	no
connect	no	no	yes
isConnect	no	no	yes
disconnect	no	no	yes
timeout or connect	yes	yes	no
verify connection interval	yes	no	no

Controlling When a Connection is Made

By using Connector Settings, you may have e*Way connections controlled manually (through the Collaboration) or automatically (through the e*Way Connection Configuration). If you choose to control the connection manually, you can have the e*Way Connection made:

- When the Collaboration is loaded
- When the Collaboration is executed
- By using an additional connection method in the ETD
- By overriding any custom values you have assigned in the Collaboration

- By using the `isConnected()` method (if your ETD has multiple connections, the `isConnected()` method is called *per connection*)

Controlling When a Connection is Disconnected

In addition to controlling when a connection is made, you can also control when an e*Way Connection is terminated or disconnected either manually or automatically. You can have the e*Way Connection disconnect:

- At the end of a Collaboration
- At the end of the execution of the Collaborations Business Rules
- During a timeout
- After a method call

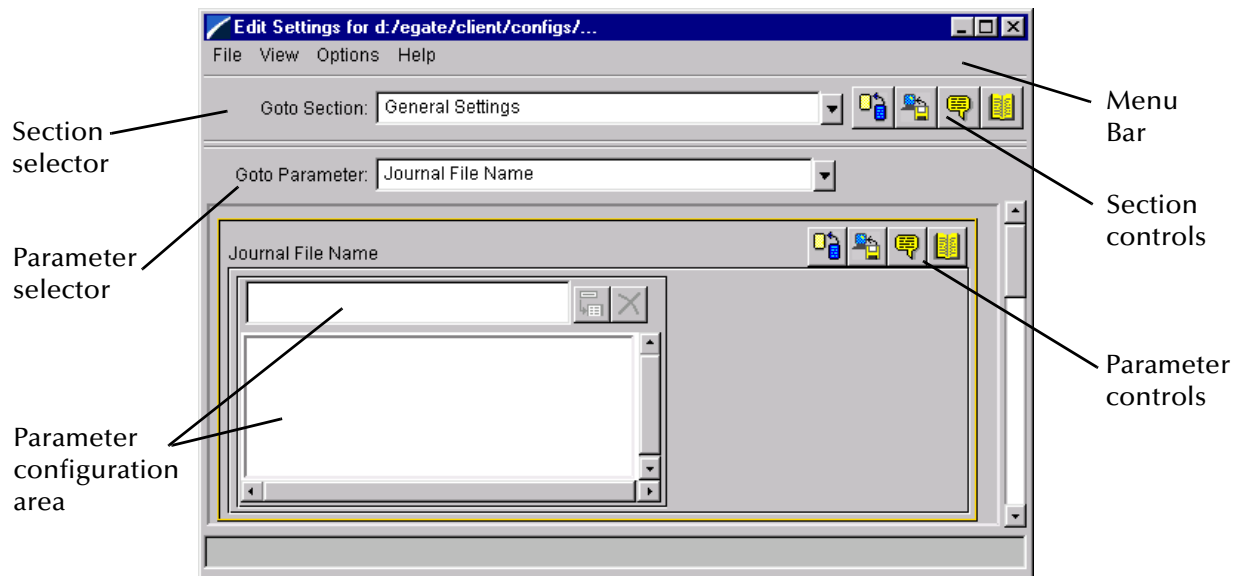
Controlling the Connectivity Status

You can control how often the e*Way Connection checks to verify is still live (see [Connector Settings](#) on page 106).

5.5 Using the e*Way Configuration Editor

The e*Way's default configuration parameters are stored in an ASCII text file with a **.def** extension. The e*Way Editor provides a simple graphical interface for viewing and changing those parameters to create a working configuration file pair (**.cfg** and **.sc**).

Figure 37 The e*Way Configuration Editor







The e*Way Configuration Editor controls fall into one of six categories:

- The **Menu bar** allows access to basic operations (e.g., saving the configuration file, viewing a summary of all parameter settings, and launching the Help system)
- The **Section selector** at the top of the Editor window enables you to select the category of the parameters you wish to edit
- **Section controls** enable you to restore the default settings, restore the last saved settings, display tips, or enter comments for the currently selected section
- The **Parameter selector** allows you to jump to a specific parameter within the section, rather than scrolling
- **Parameter controls** enable you to restore the default settings, restore the last saved settings, display tips, or enter comments for the currently selected parameter
- **Parameter configuration controls** enable you to set the e*Way's various operating parameters

5.5.1 Section and Parameter Controls

The section and parameter controls are shown in Table 8 below.

Table 8 Parameter and Section Controls

Button	Name	Function
	Restore Default	Restores default values
	Restore Value	Restores saved values
	Tips	Displays tips
	User Notes	Enters user notes



Note: The section controls affect all parameters in the selected section, whereas the parameter controls affect only the selected parameter.

5.5.2 Parameter Configuration Controls

Parameter configuration controls fall into one of two categories:

- Option buttons
- Selection lists, which have controls as described in Table 9

Table 9 Selection List Controls

Button	Name	Function
	Add to List	Adds the value in the text box to the list of available values.
	Delete Items	Displays a “delete items” dialog box, used to delete items from the list.

5.5.3 File Menu Selections

- **Save**
 - ♦ If you are editing a **.def** file, selecting **Save** creates a working configuration file pair (**.cfg** and **.sc**) from the file you are editing.
 - ♦ If you are editing an existing **.cfg** file, selecting **Save** overwrites the existing configuration file pair (**.cfg** and **.sc**).
- **Promote to Run Time**
 - ♦ Selecting **Promote to Run Time** promotes the configuration file pair (**.cfg** and **.sc**) from the Sandbox to the registry, committing them to the current schema.

For more information, see the *Getting Started* chapter of the *e*Gate Integrator User's Guide*.

5.5.4 Command-line Configuration

In the **Additional Command Line Arguments** box, type any additional command line arguments that the e*Way may require, taking care to insert them *at the end* of the existing command-line string. Be careful not to change any of the default arguments unless you have a specific need to do so.

5.5.5 Getting Help

To launch the e*Way Editor's Help system

From the **Help** menu, select **Help topics**.

To display tips regarding the general operation of the e*Way

From the **File** menu, select **Tips**.

To display tips regarding the selected Configuration Section

In the **Section Control** group, click .

To display tips regarding the selected Configuration Parameter

In the **Parameter Control** group, click .

Note: *Tips are displayed and managed separately from the online Help system. You cannot search for Tips within the Help system, or view Help system topics by requesting Tips.*

For detailed descriptions and procedures for using the e*Way Configuration Editor, see the *Working with e*Ways* chapter of the *e*Gate Integrator User's Guide*.

5.6 Troubleshooting the e*Way

In the initial stages of developing your e*Gate Integrator system administration system, most problems with e*Ways can be traced to configuration.

5.6.1 Configuration Problems

In the Schema Designer

- Does the e*Way have the correct Collaborations assigned?
- Do those Collaborations use the correct Collaboration Services?
- Is the logic correct within any Collaboration Rules script employed by this e*Way's Collaborations?
- Do those Collaborations subscribe to and publish Events appropriately?
- Are all the components that provide information to this e*Way properly configured, and are they sending the appropriate Events correctly?
- Are all the components to which this e*Way sends information properly configured, and are they subscribing to the appropriate Events correctly?

In the e*Way Editor

- Check that all e*Way connection options are set appropriately.
- Check that all settings you changed are set correctly.
- Check all required changes to ensure they have not been overlooked.
- Check the defaults to ensure they are acceptable for your installation.

On the e*Way's Participating Host

- Check that the Participating Host is operating properly, and that it has sufficient disk space to hold the IQ data that this e*Way's Collaborations publish.
- Check that your *path* environmental variable includes the location of the Oracle Manufacturing dynamically-loaded libraries. The name of this variable on the different operating systems is:
 - ♦ PATH (Windows)
 - ♦ LD_LIBRARY_PATH (Solaris)
 - ♦ LIBPATH (AIX)
 - ♦ SHLIB_PATH (HP-UX)
- Check that your *classpath* environmental variable includes the location of the Oracle Manufacturing e*Way Connection Java classes. The name of this variable can vary.

In the External Application

- Check that the application is configured correctly, is operating properly, and is sending or receiving the correct data appropriately.

5.6.2 System-related Problems

- Check that the connection between the external application and the e*Way is functioning appropriately.
- Once the e*Way is up and running properly, operational problems can be due to:
 - ♦ External influences (network or other connectivity problems).
 - ♦ Problems in the operating environment (low disk space or system errors)
 - ♦ Problems or changes in the data the e*Way is processing.
 - ♦ Corrections required to Collaboration Rules scripts that become evident in the course of normal operations.

One of the most important tools in the troubleshooter's arsenal is the e*Way log file. See the *e*Gate Integrator Alert and Log File Reference Guide* for an extensive explanation of log files, debugging options, and using the Schema Manager to monitor operations and performance.

Operational Overview

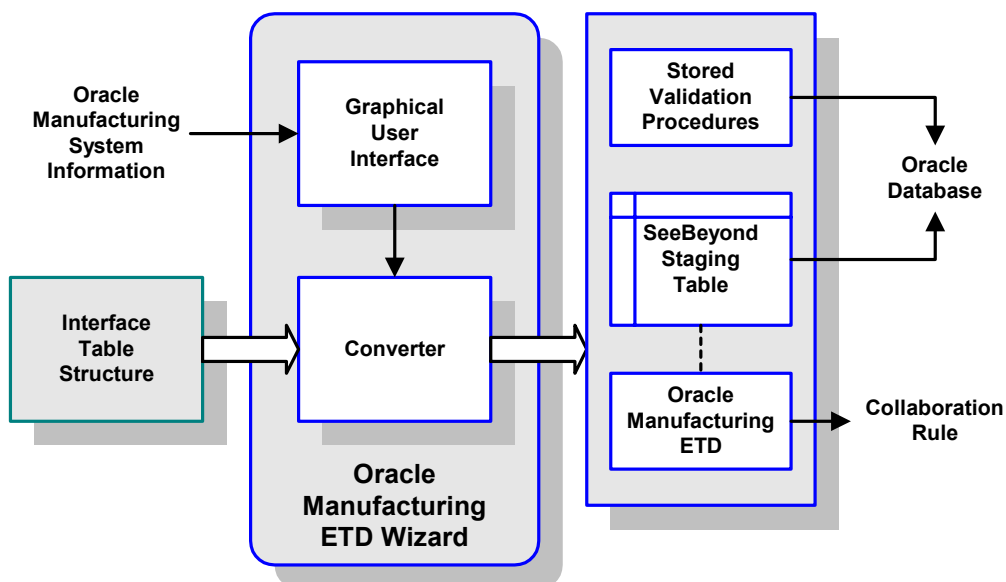
This chapter provides an overview of the basic functionality of the SeeBeyond e*Way Intelligent Adapter for Oracle Manufacturing.

6.1 Interaction With Oracle

6.1.1 Generating an ETD

The Oracle Manufacturing e*Way incorporates an ETD Wizard that helps customize the e*Way to your Oracle Manufacturing system. The operation of the Wizard is depicted in Figure 38.

Figure 38 Oracle Manufacturing ETD Wizard



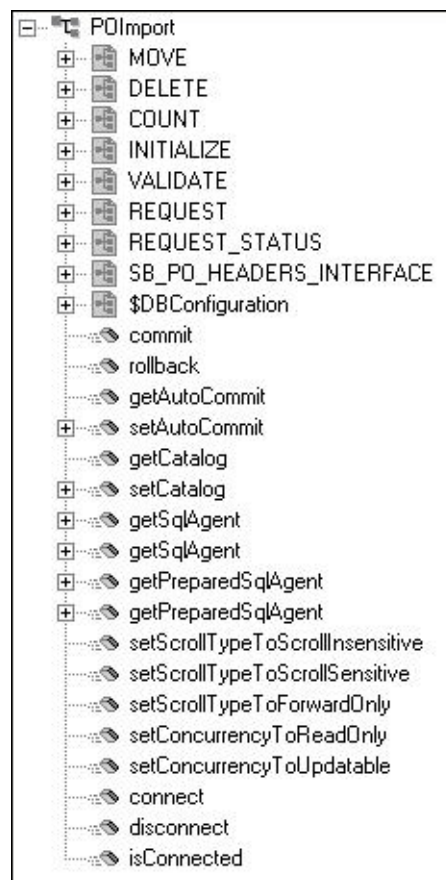
After you enter information regarding the Oracle Manufacturing system into the Wizard's GUI, the Wizard queries the specified XML configuration file to obtain the Open Interface Table hierarchy. The Wizard uses this structural metadata to build Event Type Definitions (ETDs) that are used in Collaborations, which transform Events to Oracle's data structure.

The Wizard also creates a staging table for pre-validation data and a set of stored procedures, both of which are maintained in the Oracle database. The staging table mirrors the Open Interface Table structure, with the addition of error and ID columns. The stored procedures are used in the pre-validation process.

Note: While the e*Way provides the framework, it is up to you to insert the data into the staging tables, run the pre-validations, copy over valid records, export errors, and so on. The ETD generated by the Wizard contains methods to handle most of these operations, but you must provide the logic in the Collaboration to implement them.

Figure 39 shows an example ETD where <interface> = PO_HEADERS_INTERFACE.

Figure 39 Example ETD



6.1.2 Exposed ETD Nodes

Staging Table Node

This node represents the SeeBeyond Staging Table created inside the Oracle database. All columns in the table are exposed, and can be dragged and dropped in the Collaboration Editor GUI. The node has a name of the form

SB_<Oracle Interface Table name>

having a maximum length of 30 characters.

COUNT

Description

Stored procedure defined in the utility package.

- If it is located at the ETD root level, it will count the data from *all* Staging Tables.
- If it is located at the Staging Table level, then it will count the data only for that specific Staging Table.

Parameters

Depends upon specific implementation. Typically, it contains at least four input VARCHAR parameters corresponding to:

- **sb_eway_id**
- **sb_group_id**
- **sb_object_id**

Note: If you do not assign a value (including the null value) to the above parameters, the procedure acts on all associated records.

- **sb_pass_or_fail**

This parameter accepts the following values:

- ♦ P (records that have passed)
- ♦ F (records that have failed)
- ♦ A (all records)

Requirements

The stored procedure name is derived from the Open Interface name or the Staging Table name, according to the following convention:

- If located at the root level:

SB_CNT_<OPEN INTERFACE NAME>

- If located at the staging table level:

SB_CNT_<SHORT TABLE NAME>

DELETE

Description

Stored procedure defined in the utility package.

- If it is located at the ETD root level, it will delete the data from *all* Staging Tables.
- If it is located at the Staging Table level, then it will delete the data only for that specific Staging Table.

Parameters

Depends upon specific implementation. Typically, it contains at least four input VARCHAR parameters corresponding to:

- `sb_eway_id`
- `sb_group_id`
- `sb_object_id`

Note: *If you do not assign a value (including the null value) to the above parameters, the procedure acts on all associated records.*

- `sb_pass_or_fail`

This parameter accepts the following values:

- ♦ P (records that have passed)
- ♦ F (records that have failed)
- ♦ A (all records)

Requirements

The stored procedure name is derived from the Open Interface name or the Staging Table name, according to the following convention:

- If located at the root level:

`SB_DEL_<OPEN INTERFACE NAME>`

- If located at the staging table level:

`SB_DEL_<SHORT TABLE NAME>`

INITIALIZE

Description

This optional, packaged stored procedure is used to initialize the user's profile for Oracle Manufacturing.

Parameters

Depends upon specific implementation. Typically, it takes the Organization ID as a parameter.

Requirements

Inside the script package, this stored procedure must have the name **Initialize_Profile**.

MOVE

Description

Stored procedure defined in the utility package.

- If it is located at the ETD root level, it will copy the data from *all* SeeBeyond Staging Tables to the corresponding Oracle Interface Tables.
- If it is located at the Interface Table level, then it will copy the data for only that specific Staging Table to its corresponding Open Interface Table.

This procedure acts only on records with the ID values specified.

Parameters

Depends upon specific implementation. Typically, it contains at least four input VARCHAR parameters corresponding to:

- `sb_eway_id`
- `sb_group_id`
- `sb_object_id`
- `sb_pass_or_fail`

This parameter accepts the following values:

- ♦ P (records that have passed)
- ♦ F (records that have failed)
- ♦ A (all records)

Requirements

The stored procedure name is derived from the Open Interface name or the Staging Table name, according to the following convention:

- If located at the root level:

`SB_MOV_<OPEN INTERFACE NAME>`

- If located at the staging table level:

`SB_MOVE_<SHORT TABLE NAME>`

REQUEST

Description

Concurrent Manager request function. This function is used to submit the concurrent management request to Oracle Manufacturing.

Parameters

Depends upon specific implementation.

Requirements

The function name is derived from the Open Interface name specified in the XML, and has the form: FN_REQUEST_<ORACLE INTERFACE NAME>.

REQUEST_STATUS

Description

Function used to retrieve the status of the Concurrent Manager request.

Parameters

- **INp_request_id IN NUMBER**
Request Id for the concurrent Manager; basically, the return value from **REQUEST**.
- **INp_interval_sec IN NUMBER**
The interval in seconds for the program to query for the result of a Concurrent Manager request.
- **INp_maximum_sec IN NUMBER**
The maximum allowed interval (in seconds) for the program to time out. This parameter *must* have a non-zero value.
- **OUTp_detailed_status OUT VARCHAR2**
Output parameter having the detailed description of the concurrent request.

Requirements

In order for **Request_Status** to correctly retrieve the Concurrent Manager request, you must call **commit** after the **REQUEST** stored procedure call; otherwise, **Request_Status** will always return **Pending** status after a timeout.

VALIDATE

Description

This packaged stored procedure is used to perform the pre-validation of data in the Staging Table.

- If it is located at the ETD root level, it will validate the data in *all* Staging Tables.
- If it is located at the Staging Table level, it will validate the data only in that specific Staging Table.

Parameters

Depends upon specific implementation. By default, it contains three input VARCHAR parameters corresponding to:

- `sb_eway_id`
- `sb_group_id`
- `sb_object_id`

Note: *If you do not assign a value (including the null value) to the above parameters, the procedure acts on all associated records.*

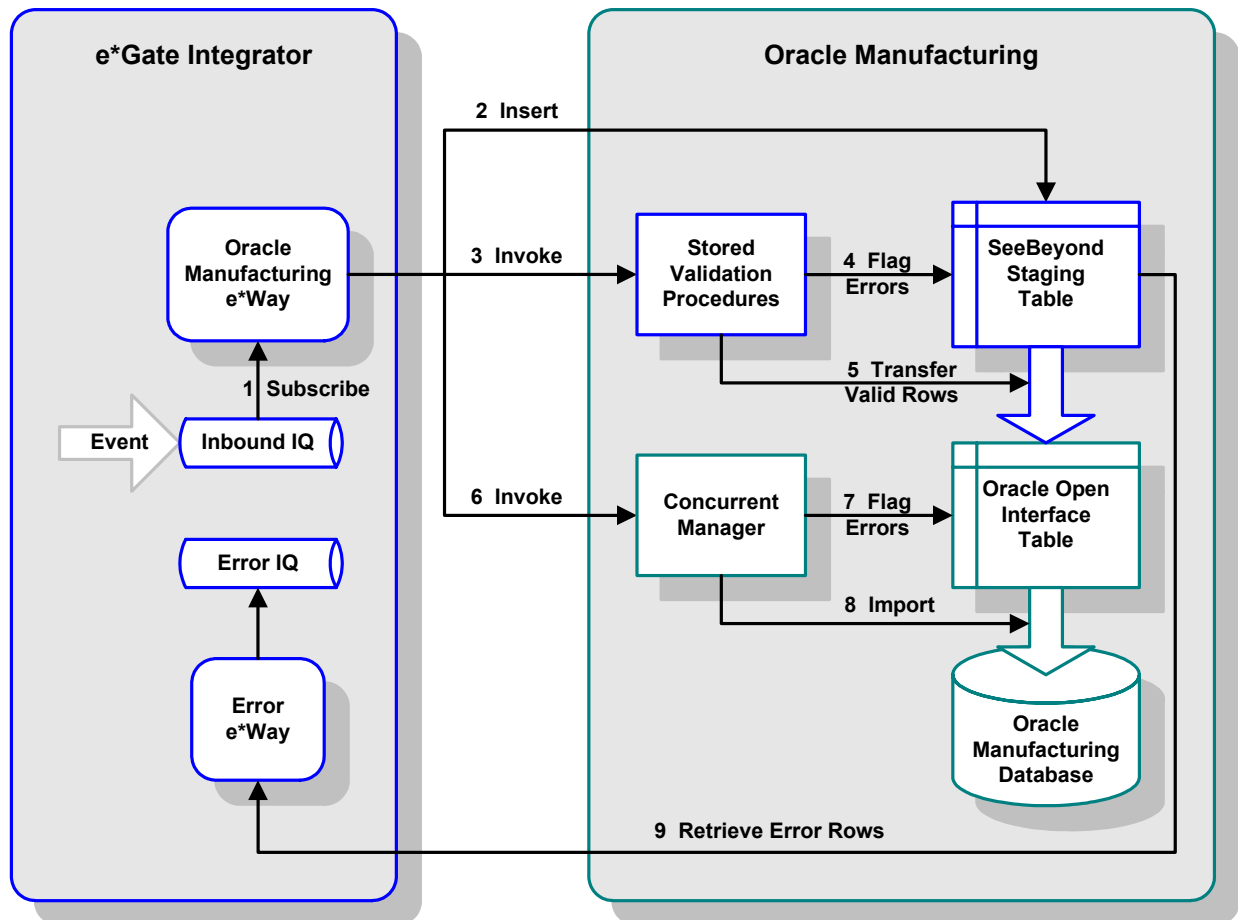
Requirements

Inside the script package, this stored procedure must have the name **Validate**.

6.1.3 Transmitting Data

Figure 40 depicts the procedure followed to insert or update the Oracle Manufacturing database with an Event from some external system. The individual steps are described below.

Figure 40 e*Gate-to-Oracle Operation



- 1 A Collaboration in the Oracle Manufacturing e*Way subscribes to data, coming from some other system via e*Gate Integrator, and transforms the data into the proper format for inserting into the relevant Open Interface Table (OIT).
- 2 The Collaboration Rule inserts the data into a SeeBeyond Staging Table, corresponding to the relevant Open Interface Table (OIT). The SeeBeyond Staging Table mirrors the Open Interface Table structure, plus additional columns for the pre-validation status and ID values.
- 3 The Collaboration Rule calls a SeeBeyond stored procedure to pre-validate the rows in the SeeBeyond Staging Table.
- 4 The stored procedure executes whatever queries are required to pre-validate the data, and updates the specific error-code columns in the SeeBeyond Staging Table. When complete, each row in the staging table has been marked with a status

indicating whether or not it passed pre-validation. If not, the row's error code indicates the type of error, with an explanation.

- 5 Depending on the business rules for the process, the e*Way invokes a number of stored procedures to handle the rows in the SeeBeyond Staging Table (there is one set of such stored procedures per SeeBeyond Staging Table).

For example, depending on the business requirements,

- ♦ It could delete all invalid rows before proceeding to move valid rows to the Open Interface Table
 - ♦ It could publish all invalid rows to another system (or allow them to be routed there by another e*Way)
 - ♦ It could abort on finding one or more errors (if the business requirement is that all rows succeed or all rows fail).
- 6 After the valid records are copied from the staging tables to the Open Interface tables, the e*Way invokes the relevant Oracle Concurrent Manager.
 - 7 The Concurrent Manager performs its own validation of the data in the Open Interface Table.
 - 8 The Concurrent Manager then imports the data into the appropriate Oracle Manufacturing database tables.
 - 9 Prepared statements in the ETD are used to retrieve the invalid records from the staging tables, along with the ID values for the failed rows.

6.2 Basic e*Gate/e*Way Operation

The e*Way Intelligent Adapter for Oracle Manufacturing uses Java Collaborations to interact with one or more external databases. By using the Java Collaboration Service it is possible for e*Gate components to connect to external databases and execute business rules written entirely in Java.

6.2.1 Event Type Definitions

ETDs can be categorized in two types:

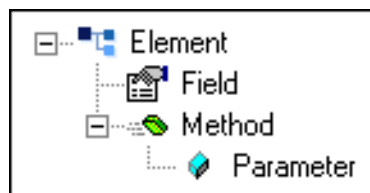
- *Data* ETDs, which represent the structure of Events (messages)
- *API-based* ETDs, or *Complex* ETDs, primarily contain APIs for communicating with external systems

Transforming data from one format to another is a major part of the processing performed by the e*Way, and data ETDs represent the data structure required by specific external systems. Generally, these ETDs are referred to as being *marshallable* because they can be *marshalled* into a flat, non hierarchical structure. See the *e*Gate Integrator User's Guide* for an extensive explanation of ETDs.

ETD Components

There are four possible components (nodes) in the Java Event Type Definition as shown in Figure 41.

Figure 41 The Java-enabled ETD

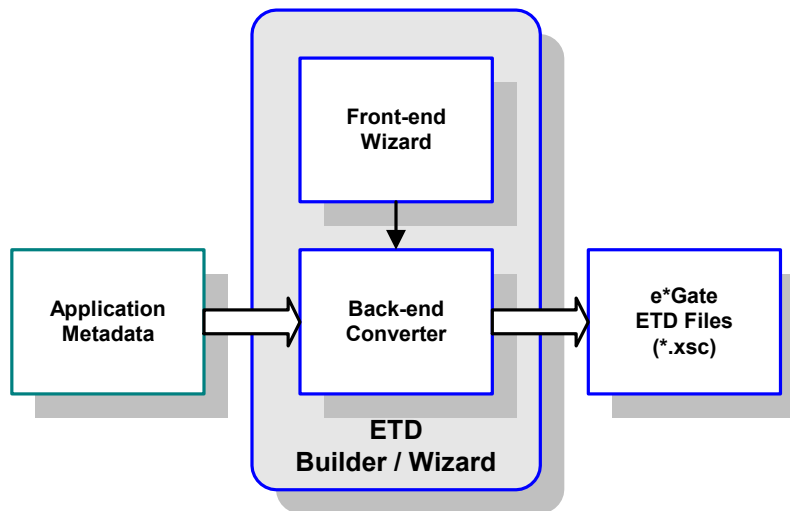


- *Elements* are the basic containers that holds the other parts of the ETD, and can contain fields and methods
- *Fields* are used to represent data, which can be in any of the following formats:
 - string
 - boolean
 - integer
 - double
 - float
- *Method* nodes represent actual Java methods
- *Parameter* nodes represent the Java method's parameters

ETD Builders

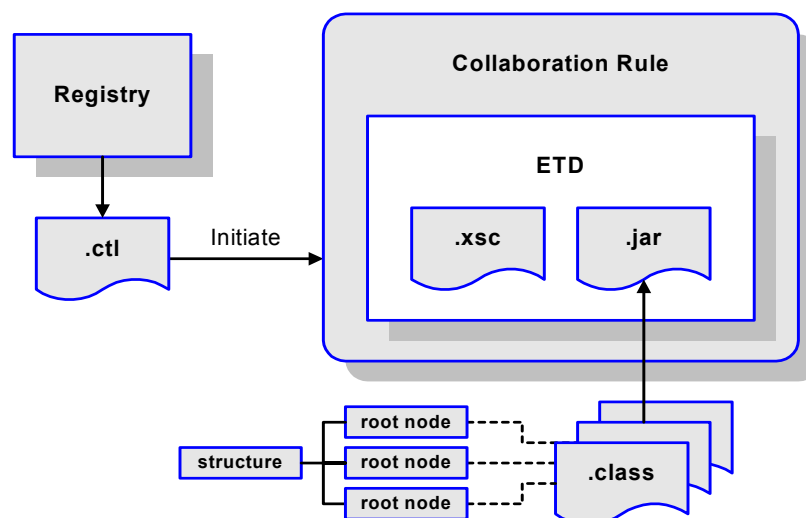
Building an ETD obviously requires knowledge of the internal data structure of the specific application. This information is obtained by extracting metadata from Oracle Manufacturing, which is automated by using the *Oracle Manufacturing ETD Wizard* (see [Figure 38](#) on page 86). For this discussion, however, consider a more typical ETD Wizard as shown in [Figure 42](#).

Figure 42 Typical ETD Wizard



Once compiled, an ETD has two components, an **.xsc** file and a **.jar** file, both having the same file name. The **.jar** file contains **.class** files whose names correspond to the root node names in the ETD. Ultimately, the ETD is used within a Collaboration Rule to define the structure of the corresponding Event. At run time, the Collaboration Rule is initiated according to information contained in a **.ctl** file contained in the e*Gate Registry (see [Figure 43](#)).

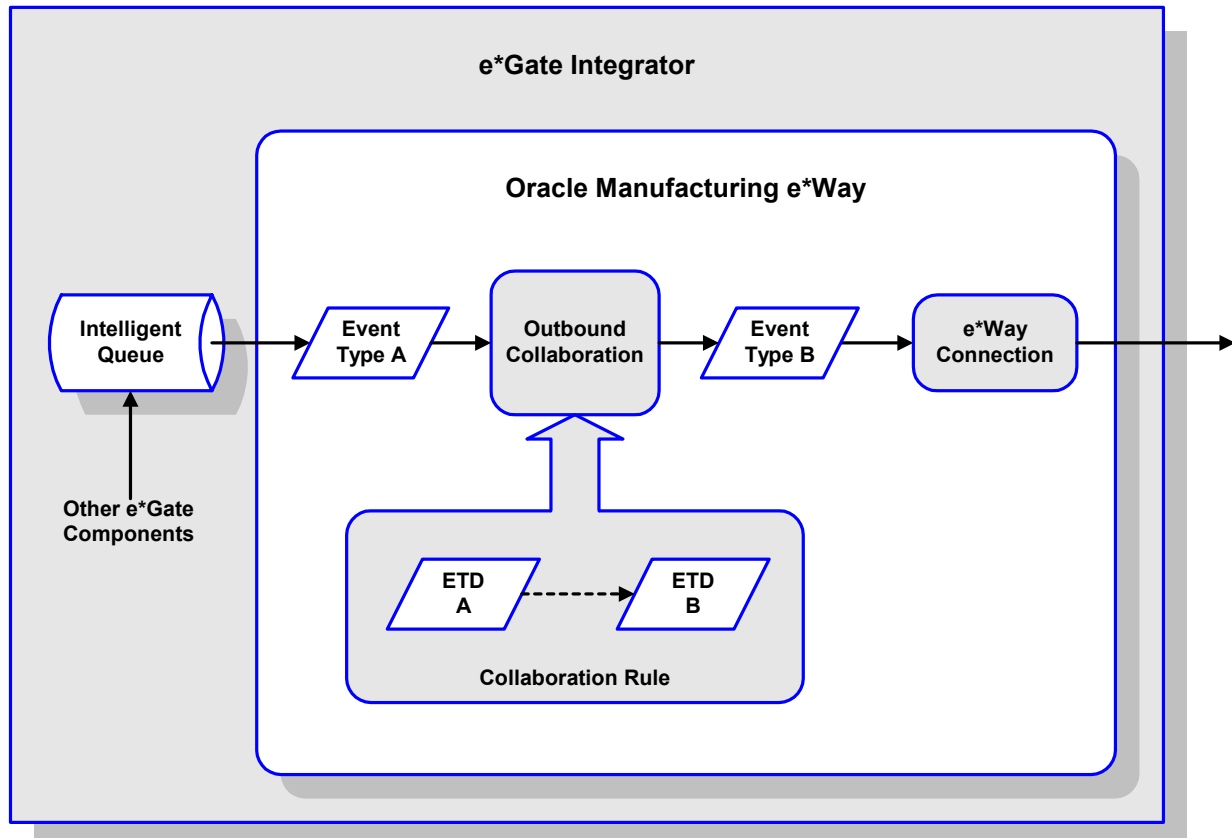
Figure 43 Event Type Definitions



6.2.2 Collaborations

Collaborations execute the business logic that enable the e*Way to perform its intended task. Each Collaboration executes a specified *Collaboration Rule*, which contains the actual instructions to execute the business logic and specifies the applicable *Event Type Definitions* (ETDs). Events Types represent *instances* of their corresponding ETDs. A look inside a typical outbound Collaboration is shown in Figure 44.

Figure 44 Outbound Collaboration

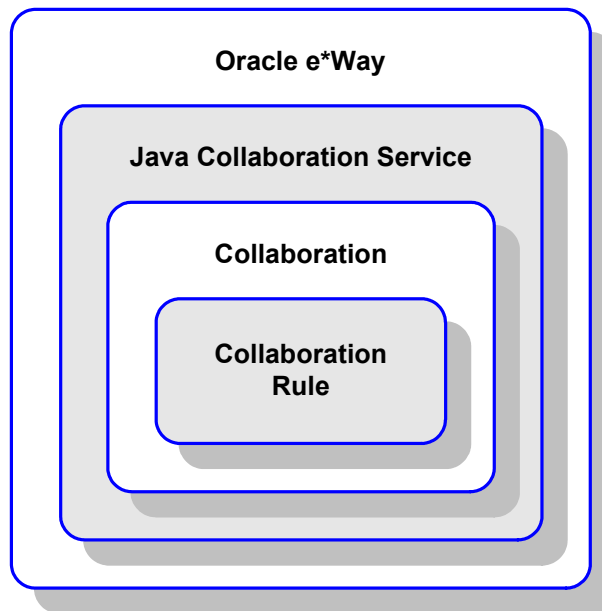


6.2.3 Java Collaboration Service

The Java Collaboration Service (JCS) provides an environment that allows you to use a Java class to implement the business logic that transforms Events as they move through e*Gate. When data passes through e*Gate using a Java Collaboration, a Java Virtual Machine (JVM) is instantiated and uses the associated Java Collaboration Rules class to accomplish the data transformation.

The relationships between the various Java e*Way components can be depicted as a nested structure, as shown in Figure 45.

Figure 45 Java Component Relationships



The Java Collaboration Service makes it possible to develop Collaboration Rules that execute e*Gate business logic using Java code. Using the Java Collaboration Editor, you create Java classes that utilize the `executeBusinessRules`, `userInitialize`, and `userTerminate` methods.

To use the Java Collaboration Service, you create a Collaboration Rule and select Java as the service. Using Event Type instances of previously defined Event Type Definitions (ETDs), you then use the Java Collaboration Rules Editor to add the rules and logic between the Event Type instances. Compiling the Collaboration Rule creates a Java Collaboration Rules class and all required supporting files. This Java class implements the data transformation logic.

For more information on the Java Collaboration Service, see the *e*Gate Integrator Collaboration Services Reference Guide*.

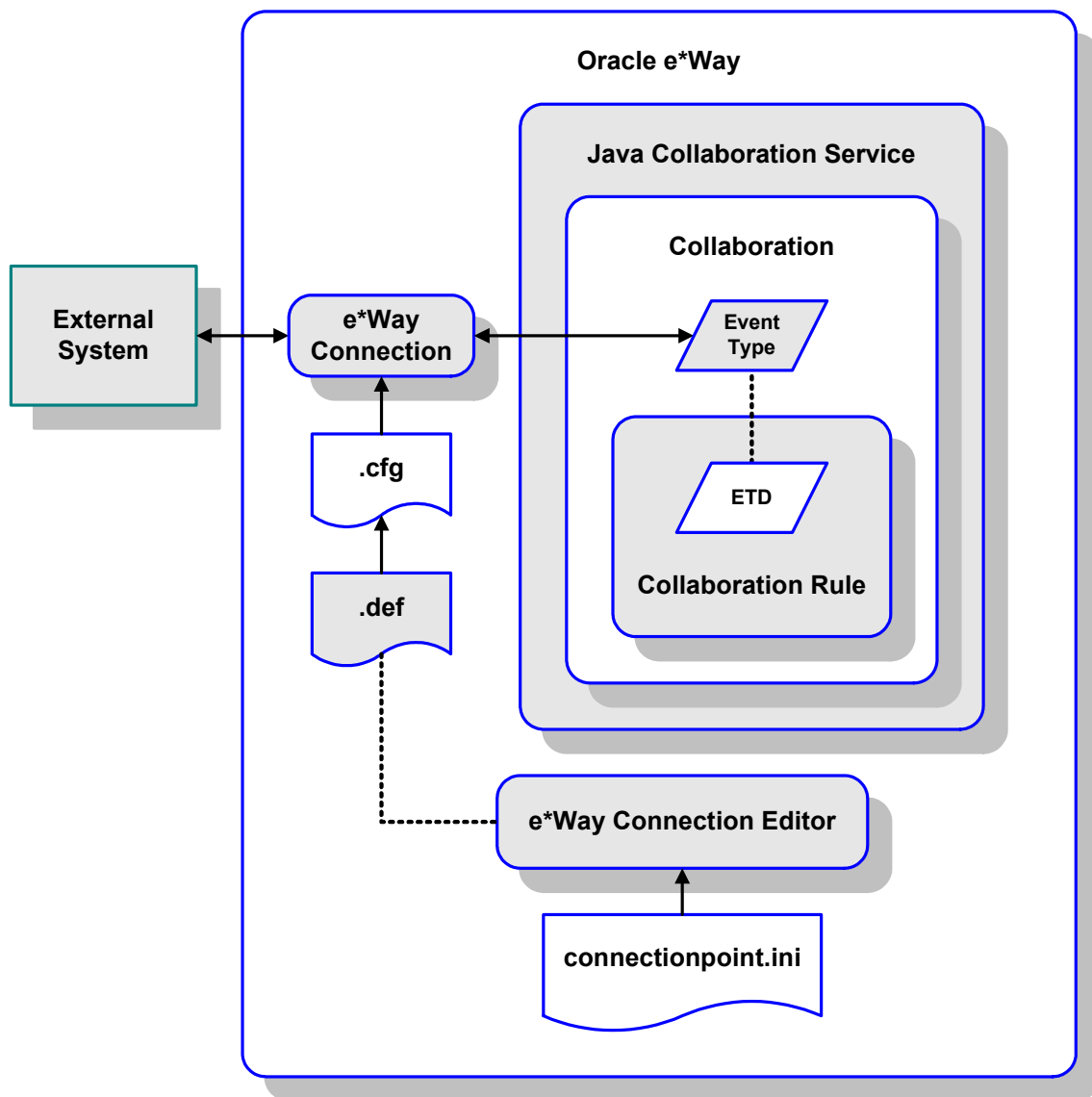
6.2.4 e*Way Connections

The e*Way Connections provide portals to external systems, allowing a single e*Way to adopt several configuration profiles simultaneously. Individual e*Way Connections can be configured using the e*Way Connection Editor to establish a particular kind of interaction with the external system.

Establishing Connections

An e*Way Connection to an external application is set up as depicted in Figure 46. The .def file supplied with the e*Way is configured for the specific application using the e*Way Connection Editor, and instantiated as a .cfg file for each e*Way Connection.

Figure 46 e*Way Connection Establishment



The e*Way Connection Editor enables you to modify all parameters of an e*Way that control the way the e*Way communicates with an external application. Because each e*Way functions in a specific way to provide an interface to a specific external application or communications protocol, each e*Way Connection has a unique set of configuration parameters.

For more information on the Java ETD Editor and the Java Collaboration Editor, see the *e*Gate Integrator User's Guide*.

Configuration Parameters

This chapter describes the configuration parameters for the Java Oracle Manufacturing e*Way.

7.1 Overview

7.1.1 e*Way Connections

An e*Way Connection's configuration parameters also are set using the e*Way Configuration Editor; see [Schema Designer Creating e*Way Connections](#) on page 76 for procedural information. The default configuration is provided in `xxx.def`. The Oracle Manufacturing e*Way's configuration parameters are organized into the following sections:

[DataSource Settings](#) on page 103

[Connector Settings](#) on page 106

7.2 e*Way Connections

7.2.1 DataSource Settings

The DataSource settings define the parameters used to interact with the external database.

class

Description

Specifies the Java class in the JDBC driver that is used to implement the **ConnectionPoolDataSource** interface.

Required Values

A valid class name.

The default is **oracle.jdbc.pool.OracleConnectionPoolDataSource**.

Additional Information

Use the **oracle.jdbc.xa.client.OracleXADataSource** class for XA compliant implementations.

To use XA in Oracle 8i and Oracle9i, your Database Administrator needs to GRANT SELECT ON DBA_PENDING_TRANSACTIONS TO PUBLIC.

Note: *XA functionality is not supported on Oracle 8.0.5 databases. For more information on implementing the Oracle e*Way in an XA compliant environment, see the e*Gate Integrator User's Guide.*

DriverType

Description

Specifies the JDBC driver type for Oracle. All other JDBC drivers are ignored.

Oracle implicitly issue a commit statement even if auto commit is set to false and no explicit commit or rollback is executed. Please see Chapter 3 of Oracle 8.1.7 JDBC Developer's Guide and Reference:

Any DDL operation, such as CREATE or ALTER, always includes an implicit COMMIT. If auto-commit mode is disabled, this implicit COMMIT will not only commit the DDL statement, but also any pending DML operations that had not yet been explicitly committed or rolled back.

Required Values

Either **thin** or **oci8**; the default is **thin**.

ServerName

Description

Specifies the host name of the external database server.

Required Values

Any valid string.

PortNumber

Description

Specifies the I/O port number on which the server is listening for connection requests.

Required Values

A valid port number. The default is 1521.

DatabaseName

Description

Specifies the name of the database instance.

Required Values

Any valid string.

user name

Description

Specifies the user name the e*Way will use to connect to the database.

Required Values

Any valid string.

password

Description

Specifies the password used to access the database.

Required Values

Any valid string.

timeout

Description

Specifies the login timeout in seconds.

Required Values

Any valid string. The default is **300** seconds.

7.2.2 Connector Settings

The Connector settings define the high level characteristics of the e*Way Connection. See [Managing e*Way Connections](#) on page 79 for additional information.

type

Description

Specifies the type of e*Way Connection. The current available type for JDBC connections is **DB**.

Required Values

The default is **DB**.

class

Description

Specifies the class name of the JDBC connector object.

Required Values

The default is **com.stc.eways.jdbcx.DbConnector**.

transaction mode

This parameter specifies how a transaction should be handled.

- **Automatic** — e*Gate will take care of transaction control and users should not issue a commit or rollback. If you are using XA, you must set your **connection establishment mode** and your **transaction mode** both to **Automatic**.
- **Manual** — You will manually take care of transaction control by issuing a commit or rollback.

Required Values

The required values are **Automatic** or **Manual**. The default is set to **Automatic**.

Mixing XA-Compliant and XA-Noncompliant e*Way Connections

A Collaboration can be XA-enabled if and only if all its sources and destinations are XA-compliant e*Way Connections. However, XA-related advantages can accrue to a Collaboration that uses one (and only one) e*Way Connection that is transactional but not XA-compliant—in other words, it connects to exactly one external system that supports commit/rollback (and is thus transactional) but does not support two-phase commit (and is thus not XA-compliant). Please see the *e*Gate User's Guide* for usage and restrictions.

connection establishment mode

This parameter specifies how a connection with the database server is established and closed.

- **Automatic** indicates that the connection is automatically established when the collaboration is started and keeps the connection alive as needed. If you are using XA, you must set your **connection establishment mode** and your **transaction mode** both to **Automatic**.
- **OnDemand** indicates that the connection will be established on demand as business rules requiring a connection to the external system are performed. The connection will be closed after the methods are completed.
- **Manual** indicates that the user will explicitly call the connection connect and disconnect methods in their collaboration as business rules.

Required Values

The required values are **Automatic**, **OnDemand** or **Manual**. The default is set to **Automatic**.

Note: If you are using Manual connection establishment mode, you must also use Manual transaction mode.

connection inactivity timeout

This value is used to specify the timeout for the Automatic connection establishment mode. If this is set to 0, the connection will not be brought down due to inactivity. The connection is always kept alive; if it goes down, re-establishing the connection will automatically be attempted. If a non-zero value is specified, the connection manager will try to monitor for inactivity so that the connection is brought down if the timeout specified is reached.

Required Values

Any valid string.

connection verification interval

This value is used to specify the minimum period of time between checks for connection status to the database server. If the connection to the server is detected to be down during verification, your collaboration's **onDown** method is called. If the connection comes up from a previous connection error, your collaboration's **onUp** method is called.

Required Values

Any valid string.

Java Classes and Methods

8.1 Overview

The Java Oracle Manufacturing e*Way contains Java methods that are used to extend the functionality of the basic e*Way core. This chapter contains descriptions of methods that are exposed in the user interface. Unless otherwise noted, all classes and methods described in this chapter are **public**.

8.2 Object Classes

The Java Oracle Manufacturing e*Way object methods are contained in the following classes:

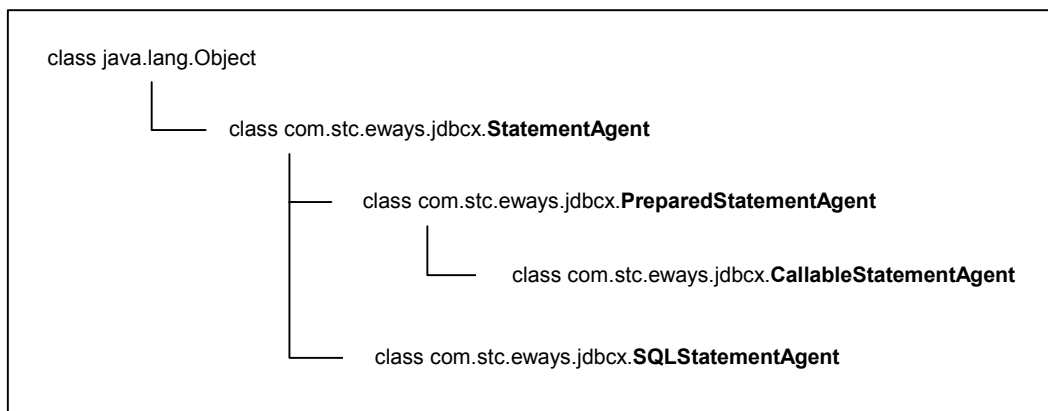
[StatementAgent Class](#) on page 109

[PreparedStatementAgent Class](#) on page 126

[CallableStatementAgent Class](#) on page 144

[SqlStatementAgent Class](#) on page 161

Figure 47 Class Hierarchy



8.3 StatementAgent Class

Description

Implements **SessionEventListener**, **ResetEventListener**

Definition

`com.stc.eways.jdbcx.StatementAgent`

Constructors

None.

Methods

[cancel](#) on page 110

[clearBatch](#) on page 110

[clearWarnings](#) on page 111

[executeBatch](#) on page 111

[getFetchDirection](#) on page 112

[getFetchSize](#) on page 112

[getMaxFieldSize](#) on page 113

[getMaxRows](#) on page 113

[getMoreResults](#) on page 114

[getResultSet](#) on page 115

[getResultSetConcurrency](#) on page 115

[getResultSetType](#) on page 116

[getUpdateCount](#) on page 116

[getWarnings](#) on page 117

[isClosed](#) on page 117

[queryDescription](#) on page 118

[queryName](#) on page 118

[resetRequested](#) on page 119

[resultSetConcurToString](#) on page 119

[resultSetDirToString](#) on page 120

[resultSetTypeToString](#) on page 120

[sessionClosed](#) on page 121

[sessionOpen](#) on page 121

[setCursorName](#) on page 122

[setEscapeProcessing](#) on page 122

[setFetchDirection](#) on page 123

[setMaxFieldSize](#) on page 124

[setMaxRows](#) on page 123

[getQueryTimeout](#) on page 114

[setQueryTimeout](#) on page 124

[stmtInvoke](#) on page 125

8.3.1 Methods

cancel

Description

This method cancels a statement that is being executed.

Signature

```
cancel ()
```

Parameters

None.

Return Type

void

Throws

java.sql.SQLException

clearBatch

Description

This method clears the batch operation.

Signature

```
clearBatch ()
```

Parameters

None.

Return Type

void

Throws

java.sql.SQLException

clearWarnings

Description

This method clears all SQL Warning objects.

Signature

```
clearWarnings()
```

Parameters

None.

Return Type

void

Throws

java.sql.SQLException

executeBatch

Description

This method executes batch statements and returns an integer array containing update counts that correspond to the commands that executed successfully. An update count of -2 indicates that the command was successful, but the number of rows affected is unknown.

Signature

```
executeBatch()
```

Parameters

None.

Return Type

integer []

Throws

java.sql.SQLException

getFetchDirection

Description

This method returns the fetch direction of the **ResultSet**: `FETCH_FORWARD`, `FETCH_REVERSE`, or `FETCH_UNKNOWN`.

Signature

```
getFetchDirection()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getFetchSize

Description

This method returns the fetch size of this **StatementAgent** object set.

Signature

```
getFetchSize()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getMaxFieldSize

Description

This method returns the maximum number of bytes that a **ResultSetAgent** column may contain; 0 indicates no limit.

Signature

```
getMaxFieldSize()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getMaxRows

Description

This method returns the maximum number of rows that a **ResultSetAgent** may contain.

Signature

```
getMaxRows()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getMoreResults

Description

This method returns an indication of whether or not there are more result sets. It returns:

- **true** if the next result is a **ResultSetAgent**
- **false** if it is an integer (indicating either an update count or that there are no more results).

Signature

```
getMoreResults()
```

Parameters

None.

Return Type

boolean

Throws

java.sql.SQLException

getQueryTimeout

Description

This method returns the number of seconds to wait before timeout.

Signature

```
getQueryTimeout()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getResultSet

Description

This method returns the **ResultSetAgent** that was produced by the call to the method `execute`.

Signature

```
getResultSet()
```

Parameters

None.

Return Type

ResultSetAgent

Throws

java.sql.SQLException

getResultSetConcurrency

Description

This method returns the result set concurrency mode.

Signature

```
getResultSetConcurrency()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getResultSetType

Description

This method returns the result set scroll type.

Signature

```
getResultSetType()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getUpdateCount

Description

This method returns the records count for the last executed statement; that is, the number of rows affected by an updated operation. It returns:

- 0 if no rows were affected or the operation was a DDL command
- -1 if the result is a **ResultSetAgent** or if there are no more results

Signature

```
getUpdateCount()
```

Parameters

None.

Return Type

integer

Throws

java.sql.SQLException

getWarnings

Description

This method returns the SQL warning object. It returns either the first SQL warning or **null** if there are no warnings.

Signature

```
getWarnings()
```

Parameters

None.

Return Type

```
java.sql.SQLWarning
```

Throws

```
java.sql.SQLException
```

isClosed

Description

This method returns the statement agent's close status; it returns **true** if the statement agent is closed.

Signature

```
isClosed()
```

Parameters

None.

Return Type

```
boolean
```

Throws

None.

queryDescription

Description

This method returns the description of the query.

Signature

```
queryDescription()
```

Parameters

None.

Return Type

java.lang.String

Throws

None.

queryName

Description

This method, specified by **queryName** in the interface **SessionEventListener**, returns the class name of the listener.

Signature

```
queryName()
```

Parameters

None.

Return Type

java.lang.String

Throws

None.

resetRequested

Description

This method, specified by **resetRequested** in the interface **ResetEventListener**, resets the event handler.

Signature

```
resetRequested(evt)
```

Parameters

Name	Type	Description
evt	ResetEvent	The requested Reset Event.

Return Type

void

Throws

java.sql.SQLException

resultSetConcurToString

Description

This method returns the symbol string corresponding to the **ResultSet** concurrency enumeration.

Signature

```
resultSetConcurToString(concur)
```

Parameters

Name	Type	Description
concur	integer	The ResultSet concurrency.

Return Type

java.lang.String

Throws

None.

resultSetDirToString

Description

This method returns the symbol string corresponding to the **ResultSet** direction enumeration.

Signature

```
resultSetDirToString(dir)
```

Parameters

Name	Type	Description
dir	integer	The ResultSet scroll direction.

Return Type

java.lang.String

Throws

None.

resultSetTypeToString

Description

This method returns the symbol string corresponding to the **ResultSet** type enumeration.

Signature

```
resultSetTypeToString(type)
```

Parameters

Name	Type	Description
type	integer	The ResultSet type.

Return Type

java.lang.String

Throws

None.

sessionClosed

Description

This method, specified by **sessionClosed** in the interface **SessionEventListener**, closes the session event handler.

Signature

```
sessionClosed(evt)
```

Parameters

Name	Type	Description
evt	SessionEvent	Session event.

Return Type

void

Throws

None.

sessionOpen

Description

This method, specified by **sessionOpen** in the interface **SessionEventListener**, opens the session event handler.

Signature

```
sessionOpen(evt)
```

Parameters

Name	Type	Description
evt	SessionEvent	Session event.

Return Type

void

Throws

None.

setCursorName

Description

This method sets the result set cursor name to the value specified.

Signature

```
setCursorName (sName)
```

Parameters

Name	Type	Description
sName	java.lang.String	The value for the Cursor name field.

Return Type

void

Throws

java.sql.SQLException

setEscapeProcessing

Description

This method sets the escape syntax processing.

Signature

```
setEscapeProcessing (bEscape)
```

Parameters

Name	Type	Description
bEscape	boolean	True to enable, False to disable.

Return Type

void

Throws

java.sql.SQLException

setFetchDirection

Description

This method sets the result set fetch direction.

Signature

```
setFetchDirection (iDir)
```

Parameters

Name	Type	Description
iDir	integer	The fetch direction of the ResultSet: FETCH_FORWARD, FETCH_REVERSE, or FETCH_UNKNOWN.

Return Type

void

Throws

java.sql.SQLException

setMaxRows

Description

This method sets the maximum number of fetch records.

Signature

```
setMaxRows (nRow)
```

Parameters

Name	Type	Description
nRow	integer	The maximum number of rows allowed in the ResultSetAgent.

Return Type

void

Throws

java.sql.SQLException

setQueryTimeout

Description

This method sets the value of the query timeout duration.

Signature

```
setQueryTimeout(nInterval)
```

Parameters

Name	Type	Description
nInterval	integer	The number of seconds before timeout.

Return Type

void

Throws

java.sql.SQLException

setMaxFieldSize

Description

This method sets the maximum field data size.

Signature

```
setMaxFieldSize (nSize)
```

Parameters

Name	Type	Description
nSize	int	The maximum size allowed for a column in a ResultSetAgent.

Return Type

void

Throws

java.sql.SQLException

stmtInvoke

Description

This method invokes a method of the database Statement object of the current ETD. It returns the Object instance resulting from the method invocation, and can be **null** if nothing is returned (void return declaration).

Signature

```
stmtInvoke (methodName, argsCls, args)
```

Parameters

Name	Type	Description
methodName	java.lang.String	The name of the method.
argsCls	java.lang.Class []	Class array for types of formal arguments for method, in the declared order. Can be null if there are no formal arguments. However, cannot invoke constructor here.
args	java.lang.Object []	Object array of formal arguments for method in the declared order. Can be null if there are no formal arguments. However, cannot invoke constructor here.

Return Type

java.lang.Object

Throws

java.lang.Exception, and whatever exception the invoked method throws.

8.4 PreparedStatementAgent Class

Description

This class extends the **StatementAgent** class and hosts the **PreparedStatement** Object.

Definition

```
com.stc.eways.jdbcx.PreparedStatementAgent
```

Constructors

None

Methods

addBatch on page 127	setDate on page 136
clearParameters on page 127	setDate on page 136
execute on page 127	setDouble on page 135
executeQuery on page 128	setFloat on page 134
executeUpdate on page 128	setInt on page 133
sessionOpen on page 129	setLong on page 134
setArray on page 142	setNull on page 129
setAsciiStream on page 140	setObject on page 130
setBigDecimal on page 135	setObject on page 131
setBinaryStream on page 140	setObject on page 131
setBlob on page 142	setRef on page 143
setBoolean on page 132	setShort on page 133
setByte on page 132	setString on page 139
setBytes on page 139	setTime on page 137
setCharacterStream on page 141	setTime on page 137
setClob on page 143	setTimestamp on page 138
addBatch on page 127	setTimestamp on page 138

8.4.1 Methods

addBatch

Description

This method adds a set of parameters to the list of commands to be sent as a batch.

Signature

```
addBatch()
```

Return Type

void

Throws

java.sql.SQLException

clearParameters

Description

This method clears the parameters of all values.

Signature

```
clearParameters()
```

Return Type

void

Throws

java.sql.SQLException

execute

Description

This method executes the Prepared SQL statement.

Signature

```
execute()
```

Return Type

void

Throws

java.sql.SQLException

executeQuery

Description

This method executes the prepared SQL query and returns a **ResultSetAgent** that contains the generated result set. It returns **null** if the query is unsuccessful.

Signature

```
executeQuery()
```

Return Type

ResultSetAgent

Throws

java.sql.SQLException

executeUpdate

Description

This method executes the prepared SQL statement and returns the number of rows that were affected by the update operation. It returns **0** if no rows were affected.

Signature

```
executeUpdate()
```

Return Type

integer

Throws

java.sql.SQLException

sessionOpen

Description

This method opens the session event handler.

Signature

```
sessionOpen(evt)
```

Parameters

Name	Type	Description
evt	SessionEvent	Session event.

Return Type

void

Overrides

sessionOpen in class **StatementAgent**

Throws

java.sql.SQLException

setNull

Description

This method nullifies the value of the indexed parameter.

Signature

```
setNull(index, type)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
type	integer	A JDBC type defined by <code>inava.sql.Types</code>

Return Type

void

Throws

java.sql.SQLException

setNull

Description

This method nullifies value of indexed parameter.

Signature

```
setNul(index, type, tname)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
type	integer	A JDBC type defined by inava.sql.Types
tname	java.lang.String	The fully-qualified name of the parameter being set. If type is not REF, STRUCT, DISTINCT, or JAVA_OBJECT, this parameter will be ignored.

Return Type

void

Throws

java.sql.SQLException

setObject

Description

This method sets the value of indexed parameter with an Object.

Signature

```
setObject(index, ob)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
ob	java.lang.Object	An instance of a Java Object containing the input parameter value.

Return Type

void

Throws

java.sql.SQLException

setObject

Description

This method sets the value of the indexed parameter with an Object.

Signature

```
setObject(index, ob, iType)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
ob	java.lang.Object	An instance of a Java Object containing the input parameter value.
iType	integer	A JDBC type defined by inava.sql.Types

Return Type

void

Throws

java.sql.SQLException

setObject

Description

This method sets the value of the indexed parameter with an Object.

Signature

```
setObject(index, ob, iType, iScale)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
ob	java.lang.Object	An instance of a Java Object containing the input parameter value.
iType	integer	A JDBC type defined by inava.sql.Types
iScale	integer	The number of digits to the right of the decimal point. Only applied to DECIMAL and NUMERIC types

Return Type

void

Throws

java.sql.SQLException

setBoolean

Description

This method sets the boolean value of the indexed parameter.

Signature

```
setBoolean(index, b)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
b	boolean	Either true or false.

Return Type

void

Throws

java.sql.SQLException

setByte

Description

This method sets the byte value of the indexed parameter.

Signature

```
setByte(index, byt)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
byt	byte	The byte parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setShort

Description

This method sets the short value of the indexed parameter.

Signature

```
setShort(index, si)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
si	short	The short parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setInt

Description

This method sets the integer value of the indexed parameter.

Signature

```
setInt(index, i)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
i	integer	The integer parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setLong

Description

This method sets the long value of the indexed parameter.

Signature

```
setLong(index, l)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
l	long	The long parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setFloat

Description

This method sets the float value of the indexed parameter.

Signature

```
setFloat(index, f)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
f	float	The float parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setDouble

Description

This method sets the double value of the indexed parameter.

Signature

```
setDouble(index, d)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
d	double	The double parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setBigDecimal

Description

This method sets the decimal value of the indexed parameter.

Signature

```
setBigDecimal(index, dec)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
dec	java.math.BigDecimal	The BigDecimal parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setDate

Description

This method sets the date value of the indexed parameter.

Signature

```
setDate(index, date)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
date	java.sql.Date	The Date parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setDate

Description

This method sets the date value of the indexed parameter with the time zone from the calendar.

Signature

```
setDate(index, date, cal)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
date	java.sql.Date	The Date parameter value to be set.
cal	java.util.Calendar	The calendar object used to construct the date.

Return Type

void

Throws

java.sql.SQLException

setTime

Description

This method sets the time value of the indexed parameter.

Signature

```
setTime(index, t)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
t	java.sql.Time	The Time parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setTime

Description

This method sets the time value of the indexed parameter with the time zone from the calendar.

Signature

```
setTime(index, t, cal)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
t	java.sql.Time	The Time parameter value to be set.
cal	java.util.Calendar	The Calendar object used to construct the time.

Return Type

void

Throws

java.sql.SQLException

setTimestamp

Description

This method sets the timestamp value of the indexed parameter.

Signature

```
setTimestamp(index, ts)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
ts	java.sql.Timestamp	The Timestamp parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setTimestamp

Description

This method sets the timestamp value of the indexed parameter with the time zone from the calendar.

Signature

```
setTimestamp(index, ts, cal)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
ts	java.sql.timestamp	The Timestamp parameter value to be set.
cal	java.util.Calendar	The Calendar object used to construct the timestamp.

Return Type

void

Throws

java.sql.SQLException

setString

Description

This method sets the string value of the indexed parameter.

Signature

```
setString(index, s)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
s	java.lang.String	The String parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setBytes

Description

This method sets the byte array value of the indexed parameter.

Signature

```
setBytes(index, bytes)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
bytes	byte[]	The byte array parameter value to be set.

Return Type

void

Throws

java.sql.SQLException

setAsciiStream

Description

This method sets the character value of the indexed parameter with an input stream of specified length.

Signature

```
setAsciiStream(index, is, length)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
is	java.io.InputStream	The InputStream that contains the Ascii parameter value to be set.
length	integer	The number of bytes to be read from the stream and sent to the database.

Return Type

void

Throws

java.sql.SQLException

setBinaryStream

Description

This method sets the binary value of the indexed parameter with an input stream of specified length.

Signature

```
setBinaryStream(index, is, length)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
is	java.io.InputStream	The InputStream that contains the binary parameter value to be set.
length	integer	The number of bytes to be read from the stream and sent to the database.

Return Type

void

Throws

java.sql.SQLException

setCharacterStream

Description

This method sets the character value of the indexed parameter with a reader stream of specified length.

Signature

```
setCharacterStream(index, rd, length)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
rd	java.io.Reader	The Reader that contains the Unicode parameter value to be set.
length	integer	The number of characters to be read from the stream and sent to the database.

Return Type

void

Throws

java.sql.SQLException

setArray

Description

This method sets the array value of the indexed parameter.

Signature

```
setArray(index, a)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
a	java.sql.Array	The Array value to be set.

Return Type

void

Overrides

None.

Throws

java.sql.SQLException

setBlob

Description

This method sets the **blob** value of the indexed parameter.

Signature

```
setBlob(index, blob)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
blob	java.sql.Blob	The Blob value to be set.

Return Type

void

Overrides

None.

Throws

java.sql.SQLException

setClob

Description

This method sets the **clob** value of the indexed parameter.

Signature

```
setClob(index, clob)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
clob	java.sql.Clob	The Clob value to be set.

Return Type

void

Overrides

None.

Throws

java.sql.SQLException

setRef

Description

This method sets the **ref** value of the indexed parameter.

Signature

```
setRef(int index, ref)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
ref	java.sql.Ref	The Ref parameter value to be set.

Return Type

void

Overrides

None.

Throws

java.sql.SQLException

8.5 CallableStatementAgent Class

Description

This class extends the **PreparedStatementAgent** class and hosts the **CallableStatement** interface.

Definition

```
com.stc.eways.jdbcx.CallableStatementAgent
```

Constructors

[CallableStatementAgent](#) on page 145

[CallableStatementAgent](#) on page 145

[CallableStatementAgent](#) on page 146

Methods

[getArray](#) on page 158

[getBigDecimal](#) on page 154

[getBlob](#) on page 159

[getBoolean](#) on page 150

[getByte](#) on page 151

[getBytes](#) on page 158

[getClob](#) on page 159

[getDate](#) on page 154

[getDate](#) on page 155

[getDouble](#) on page 153

[getFloat](#) on page 153

[getInt](#) on page 152

[getLong](#) on page 152

[getObject](#) on page 149

[getObject](#) on page 150

[getRef](#) on page 160

[getShort](#) on page 151

[getString](#) on page 157

[getTime](#) on page 155

[getTime](#) on page 156

[getTimeStamp](#) on page 156

[getTimeStamp](#) on page 157

[registerOutParameter](#) on page 147

[registerOutParameter](#) on page 148

[registerOutParameter](#) on page 148

[sessionOpen](#) on page 147

[wasNull](#) on page 149

8.5.1 Constructors

CallableStatementAgent

Description

This constructor creates a new **CallableStatementAgent** with scroll direction `TYPE_FORWARD_ONLY` and concurrency `CONCUR_READ_ONLY`.

Signature

```
CallableStatementAgent(session, sCommand)
```

Parameters

Name	Type	Description
<code>session</code>	Session	Connection session.
<code>sCommand</code>	<code>java.lang.String</code>	The Call statement used to invoke a stored procedure.

CallableStatementAgent

Description

This constructor creates a new **CallableStatementAgent**.

Signature

```
CallableStatementAgent(session, iScroll, iConcur)
```

Parameters

Name	Type	Description
<code>session</code>	Session	Connection session.
<code>iScroll</code>	integer	Ignored.
<code>iConcur</code>	integer	Ignored

CallableStatement Agent

Description

This constructor creates a new **CallableStatementAgent**.

Signature

```
CallableStatementAgent(session, sCommand, iScroll, iConcur)
```

Parameters

Name	Type	Description
session	Session	Connection session.
sCommand	java.lang.String	The Call statement used to invoke a stored procedure.
iScroll	integer	Scroll direction: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE, TYPE_SCROLL_SENSITIVE
iConcur	integer	Concurrency: CONCUR_READ_ONLY, CONCUR_UPDATEABLE

8.5.2 Methods

sessionOpen

Description

This method opens the session event handler.

Signature

```
sessionOpen(evt)
```

Parameters

Name	Type	Description
evt	SessionEvent	Session event.

Return Type

void

Overrides

sessionOpen in class **PreparedStatementAgent**

Throws

java.sql.SQLException

registerOutParameter

Description

This method registers the indexed OUT parameter with the specified type.

Signature

```
registerOutParameter(index, iType)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
iType	integer	A JDBC type defined by <code>inava.sql.Types</code> .

Return Type

void

Throws

java.sql.SQLException

registerOutParameter

Description

This method registers the indexed OUT parameter with the specified type and scale.

Signature

```
registerOutParameter(index, iType, iScale)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
iType	integer	A JDBC type defined by <code>inava.sql.Types</code> .
iScale	integer	The number of digits to the right of the decimal point. Only applied to DECIMAL and NUMERIC types.

Return Type

void

Throws

`java.sql.SQLException`

registerOutParameter

Description

This method registers the indexed OUT parameter with the specified user-named type or REF type.

Signature

```
registerOutParameter(index, iType, sType)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
iType	integer	A JDBC type defined by <code>inava.sql.Types</code> .
tName	<code>java.lang.String</code>	The fully-qualified name of the parameter being set. It is intended to be used by REF, STRUCT, DISTINCT, or JAVA_OBJECT.

Return Type

void

Throws

`java.sql.SQLException`

wasNull

Description

This method returns **true** if the last OUT parameter read had the value SQL NULL; otherwise, it returns **false**

Signature

```
wasNull ()
```

Parameters

None.

Return Type

boolean

Throws

java.sql.SQLException

getObject

Description

This method returns the value of the indexed parameter as an Object instance.

Signature

```
getObject (index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
java.lang.Object
```

Throws

java.sql.SQLException

getObject

Description

This method returns the value of the indexed parameter as an Object instance, and uses the parameter `map` for customer mapping of the parameter value.

Signature

```
getObject(index, map)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
map	java.util.Map	A Map object for mapping from SQL type names for user-defined types to classes in the Java programming language.

Return Type

java.lang.Object

Throws

java.sql.SQLException

getBoolean

Description

This method returns the boolean value of the indexed parameter.

Signature

```
getBoolean(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

boolean

Throws

java.sql.SQLException

getBytes

Description

This method returns the byte value of the indexed parameter.

Signature

```
getBytes(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

byte

Throws

java.sql.SQLException

getShort

Description

This method returns the short value of the indexed parameter.

Signature

```
getShort(index)
```

Parameters

None.

Return Type

short

Throws

java.sql.SQLException

getInt

Description

This method returns the integer value of the indexed parameter.

Signature

```
getInt(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

integer

Throws

java.sql.SQLException

getLong

Description

This method returns the long value of the indexed parameter.

Signature

```
getLong(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

long

Throws

java.sql.SQLException

getFloat

Description

This method returns the float value of the indexed parameter.

Signature

```
getFloat(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

float

Throws

java.sql.SQLException

getDouble

Description

This method returns the double value of the indexed parameter.

Signature

```
getDouble(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

double

Throws

java.sql.SQLException

getBigDecimal

Description

This method returns the big decimal value of the indexed parameter.

Signature

```
getBigDecimal(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
java.math.BigDecimal
```

Throws

```
java.sql.SQLException
```

getDate

Description

This method returns the date value of the indexed parameter.

Signature

```
getDate(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
java.sql.Date
```

Throws

```
java.sql.SQLException
```

getDate

Description

This method returns the date value of the indexed parameter with the calendar time zone.

Signature

```
getDate(index, calendar)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
cal	java.util.Calendar	The Calendar object used to construct the timestamp.

Return Type

java.sql.Date

Throws

java.sql.SQLException

getTime

Description

This method returns the time value of the indexed parameter.

Signature

```
getTime(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

java.sql.Time

Throws

java.sql.SQLException

getTime

Description

This method returns the time value of the indexed parameter with the calendar time zone.

Signature

```
getTime(index, calendar)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
cal	java.util.Calendar	The Calendar object used to construct the timestamp.

Return Type

java.sql.Time

Throws

java.sql.SQLException

getTimestamp

Description

This method returns the timestamp value of the indexed parameter.

Signature

```
getTimestamp(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

java.sql.Timestamp

Throws

java.sql.SQLException

getTimestamp

Description

This method returns the timestamp value of the indexed parameter.

Signature

```
getTimestamp(index, calendar)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.
cal	java.util.Calendar	The Calendar object used to construct the timestamp.

Return Type

java.sql.Timestamp

Throws

java.sql.SQLException

getString

Description

This method returns the string value of the indexed parameter.

Signature

```
getString(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

java.lang.String

Throws

java.sql.SQLException

getBytes

Description

This method returns the byte array value of the indexed parameter.

Signature

```
getBytes(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
byte[]
```

Throws

```
java.sql.SQLException
```

getArray

Description

This method returns the array value of the indexed parameter.

Signature

```
getArray(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
java.sql.Array
```

Throws

```
java.sql.SQLException
```

getBlob

Description

This method returns the **blob** value of the indexed parameter.

Signature

```
getBlob(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
java.sql.Blob
```

Throws

```
java.sql.SQLException
```

getClob

Description

This method returns the **clob** value of the indexed parameter.

Signature

```
getClob(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

```
java.sql.Clob
```

Throws

```
java.sql.SQLException
```

getRef

Description

This method returns the **Ref** value of the indexed parameter.

Signature

```
getRef(index)
```

Parameters

Name	Type	Description
index	integer	Parameter index starting from 1.

Return Type

java.sql.Ref

Throws

java.sql.SQLException

8.6 SqlStatementAgent Class

Description

This class extends the **StatementAgent** class and hosts a managed **Statement** object.

Definition

```
com.stc.eways.jdbcx.SqlStatementAgent
```

Constructors

[SqlStatementAgent](#) on page 162

[SqlStatementAgent](#) on page 162

Methods

[addBatch](#) on page 163

[execute](#) on page 163

[executeQuery](#) on page 164

[executeUpdate](#) on page 164

8.6.1 Constructors

SqlStatementAgent

Description

This constructor creates a new **SQLStatementAgent** with scroll direction `TYPE_FORWARD_ONLY` and concurrency `CONCUR_READ_ONLY`.

Signature

```
SqlStatementAgent(session)
```

Parameters

Name	Type	Description
session	Session	Connection session.

SqlStatementAgent

Description

This constructor creates a new **SQLStatementAgent**.

Signature

```
SqlStatementAgent(session, iScroll, iConcur)
```

Parameters

Name	Type	Description
session	Session	Connection session.
iScroll	integer	Scroll direction: <code>TYPE_FORWARD_ONLY</code> , <code>TYPE_SCROLL_INSENSITIVE</code> , <code>TYPE_SCROLL_SENSITIVE</code> .
iConcur	integer	Concurrency: <code>CONCUR_READ_ONLY</code> , <code>CONCUR_UPDATABLE</code> .

8.6.2 Methods

addBatch

Description

This method adds the specified SQL statement to the list of commands to be sent as a batch.

Signature

```
addBatch(sSql)
```

Parameters

Name	Type	Description
sSql	java.lang.String	SQL statement.

Return Type

void

Throws

java.sql.SQLException

execute

Description

This method executes the specified SQL statement. It returns **true** if the first result is a **ResultSetAgent**, or **false** if it is an integer.

Signature

```
execute(sSql)
```

Parameters

Name	Type	Description
sSql	java.lang.String	SQL statement.

Return Type

boolean

Throws

java.sql.SQLException

executeQuery

Description

This method executes the specified SQL query and returns either a **ResultSetAgent** that contains the generated result set, or **null**.

Signature

```
executeQuery (sSql)
```

Parameters

Name	Type	Description
sSql	java.lang.String	SQL statement.

Return Type

ResultSetAgent

Throws

java.sql.SQLException

executeUpdate

Description

This method executes the specified SQL statement and returns the number of rows that were affected by the update operation. It returns **0** if no rows were affected.

Signature

```
executeUpdate (sSql)
```

Parameters

Name	Type	Description
sSql	java.lang.String	SQL statement.

Return Type

int

Throws

java.sql.SQLException

Appendix A

Validation Scripts

A.1 Inventory

A.1.1 Customer Items

Table 10 sb_validate_cust_items_pkg

Validation Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Commodity_Active	Checks for valid values in column Commodity_Code and its status in table sb_mtl_ci_interface.	SB-INVCIT-25316
Perform_Other_Validations/ Check_Container_Org_Code	Checks for valid values in column Container_Org_Code in table sb_mtl_ci_interface.	SB-INVCIT-25319
Perform_Other_Validations/ Check_Container_Org_Id	Checks for valid values in column Container_Org_Id in table and its status sb_mtl_ci_interface.	SB-INVCIT-25317
Perform_Other_Validations/ Check_Container_Org_Name	Checks for valid values in column Container_Org_Name in table sb_mtl_ci_interface.	SB-INVCIT-25320
Perform_Other_Validations/ Check_Demand_Tolerance_Pstv	Checks for valid values in column Demand_Tolerance_Positive in table sb_mtl_ci_interface.	SB-INVCIT-25321
Perform_Other_Validations/ Check_Demand_Tolerance_Ngtv	Checks for valid values in column Demand_Tolerance_Negative in table sb_mtl_ci_interface.	SB-INVCIT-25322
Perform_Other_Validations/ Check_Dep_Plan_Build_Flag	Checks for valid values in column Departure_Plan_Build_Flag in table sb_mtl_ci_interface.	SB-INVCIT-25325
Perform_Other_Validations/ Check_Dep_Plan_Flags	Checks for valid values in combined Departure_Plan_Flags in table sb_mtl_ci_interface.	SB-INVCIT-25326
Perform_Other_Validations/ Check_Dep_Plan_Reqd_Flag	Checks for valid values in column Departure_Plan_Required_Flag in table sb_mtl_ci_interface.	SB-INVCIT-25324

Table 10 sb_validate_cust_items_pkg

Validation Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Inactive_Flag	Checks for valid values in column Inactive_Flag in table sb_mtl_ci_interface.	SB-INVCIT-25318
Perform_Other_Validations/ Check_Min_Fill_Percent	Checks for valid values in column Minimum_Fill_Percentage in table sb_mtl_ci_interface.	SB-INVCIT-25323
Validate_Foreign_Key_Checks/ Check_Address_Valid	Checks for valid values in foreign key column Address_ID in table sb_mtl_ci_interface.	SB-INVCIT-25313
Validate_Foreign_Key_Checks/ Check_Customer_Id	Checks for valid values in foreign key column Customer_Id in table sb_mtl_ci_interface.	SB-INVCIT-25310
Validate_Foreign_Key_Checks/ Check_Customer_No	Checks for valid values in foreign key column Customer_Number in table sb_mtl_ci_interface.	SB-INVCIT-25311
Validate_Foreign_Key_Checks/ Check_Customer_Name	Checks for valid values in foreign key column Customer_Name in table sb_mtl_ci_interface.	SB-INVCIT-25312
Validate_Foreign_Key_Checks/ Check_Cust_Cat_Code	Checks for valid values in foreign key column Custustomer_Category_Code in table sb_mtl_ci_interface.	SB-INVCIT-25314
Validate_Foreign_Key_Checks/ Check_Model_Cust_Item_Id	Checks for valid values in foreign key column Model_Customer_Item_Id in table sb_mtl_ci_interface.	SB-INVCIT-25315
Validate_Required_Columns/ Check_Address	Checks for the required field combination Address_Id and other address detail columns in table sb_mtl_ci_interface.	SB-INVCIT-25305
Validate_Required_Columns/ Check_Commodity	Checks for the required field combination Commodity_Code/Id columns in table sb_mtl_ci_interface.	SB-INVCIT-25307
Validate_Required_Columns/ Check_Container_Item	Checks for the required field combination Container_Item_Id/Code/Name in table sb_mtl_ci_interface.	SB-INVCIT-25308
Validate_Required_Columns/ Check_Customer	Checks for the required field combination Customer_Id/Name/Number in table sb_mtl_ci_interface.	SB-INVCIT-25304
Validate_Required_Columns/ Check_Cust_Item_Number	Checks for the required field Customer_Item_Number in table sb_mtl_ci_interface.	SB-INVCIT-25301
Validate_Required_Columns/ Check_Cust_Category	Checks for the required field combination Customer_Category/Code in table sb_mtl_ci_interface.	SB-INVCIT-25306

Table 10 sb_validate_cust_items_pkg

Validation Sub-Routine	Purpose	Error Code
Validate_Required_Columns/ Check_Item_Def_Level	Checks for the required field Item_Def_Level in table sb_mtl_ci_interface.	SB-INVCIT-25303
Validate_Required_Columns/ Check_Model_Cust_Item	Checks for the required field combination Model_Customer_Item_Id/ Number in table sb_mtl_ci_interface.	SB-INVCIT-25309
Validate_Required_Columns/ Check_Transaction_Type	Checks for the required field Transaction_Type in table sb_mtl_ci_interface.	SB-INVCIT-25302

A.1.2 Customer Item Cross-Reference

Table 11 sb_validate_ci_xrefs_pkg

Validation Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Inactive_Flag	Checks for valid values in column Inactive_Flag in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22624
Perform_Other_Validations/ Update_Fixed_Values	Checks for valid values in column Fixed_Values in table sb_mtl_ci_xrefs_interface.	N/A
Validate_Foreign_Keys/ Check_Customer_Id	Checks for valid values in foreign key column Customer_Id in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22616
Validate_Foreign_Keys/ Check_Customer_No	Checks for valid values in foreign key column Customer_No in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22617
Validate_Foreign_Keys/ Check_Customer_Name	Checks for valid values in foreign key column Customer_Name in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22618
Validate_Foreign_Keys/ Check_Address_Valid	Checks for valid values in foreign key column Address_Valid in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22619
Validate_Foreign_Keys/ Check_Cust_Cat_Code	Checks for valid values in foreign key column Cust_Cat_Code in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22620
Validate_Foreign_Keys/ Check_Customer_item_id	Checks for valid values in foreign key column Customer_item_id in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22621
Validate_Foreign_Keys/ Check_Master_Organization_id	Checks for valid values in foreign key column Master_Organization_id in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22622
Validate_Foreign_Keys/ Check_Inventory_Item_id	Checks for valid values in foreign key column Inventory_Item_id in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22623
Validate_Required_Columns/ Check_Process_flag	Checks for non-null entry in field Process_flag in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22601
Validate_Required_Columns/ Check_Process_mode	Checks for non-null entry in field Process_mode in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22602
Validate_Required_Columns/ Check_Last_update_date	Checks for non-null entry in field Last_update_date in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22603
Validate_Required_Columns/ Check_Last_updated_by	Checks for non-null entry in field Last_updated_by in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22604

Table 11 sb_validate_ci_xrefs_pkg

Validation Sub-Routine	Purpose	Error Code
Validate_Required_Columns/ Check_creation_date	Checks for non-null entry in field creation_date in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22605
Validate_Required_Columns/ Check_created_by	Checks for non-null entry in field created_by in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22606
Validate_Required_Columns/ Check_Transaction_Type	Checks for non-null entry in field Transaction_Type in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22607
Validate_Required_Columns/ Check_Cust_Item_Number	Checks for non-null entry in field Customer_Item_Number in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22608
Validate_Required_Columns/ Check_Preference_Number	Checks for non-null entry in field Preference_Number in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22609
Validate_Required_Columns/ Check_Inactive_flag	Checks for non-null entry in field Inactive_flag in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22610
Validate_Required_Columns/ Check_Lock_flag	Checks for non-null entry in field Lock_flag in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22611
Validate_Required_Columns/ Check_Cust_Category	Checks for non-null entry in field Customer_Category in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22612
Validate_Required_Columns/ Check_Address	Checks for non-null entry in field Customer_Address in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22613
Validate_Required_Columns/ Check_Item_Def_Level	Checks for non-null entry in field Item_Def_Level in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22614
Validate_Required_Columns/ Check_Customer	Checks for the required field combination Customer_Id/Name/Number in table sb_mtl_ci_xrefs_interface.	SB-INVCCR-22615

A.1.3 Cycle Count Entries

Table 12 sb_validate_cc_entries_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Delete_Flag	Checks for valid values in delete_flag in sb_mtl_cc_entries_interface table.	SB-INVCCE-23334
Perform_Other_Validations/ Check_EMPLOYEE_ID	Checks for valid values in columns employee_id in sb_mtl_cc_entries_interface table.	SB-INVCCE-23330
Perform_Other_Validations/ Check_Lock_Flag	Checks for valid values in Lock_Flag in sb_mtl_cc_entries_interface table.	SB-INVCCE-23335
Perform_Other_Validations/ Check_Process_Flag	Checks for valid values in Process_Flag in sb_mtl_cc_entries_interface table.	SB-INVCCE-23336
Perform_Other_Validations/ Check_Process_Mode	Checks for valid values in Process_Mode in sb_mtl_cc_entries_interface table.	SB-INVCCE-23337
Perform_Other_Validations/ Check_Project_ID	Checks for valid values in columns project_id in sb_mtl_cc_entries_interface table.	SB-INVCCE-23331
Perform_Other_Validations/ Check_Status_Flag	Checks for valid values in Status_Flag in sb_mtl_cc_entries_interface table.	SB-INVCCE-23338
Perform_Other_Validations/ Check_Task_ID	Checks for valid values in task_id in sb_mtl_cc_entries_interface table.	SB-INVCCE-23332
Perform_Other_Validations/ Check_Valid_Flag	Checks for valid values in valid_flag in sb_mtl_cc_entries_interface table.	SB-INVCCE-23333
Validate_Foreign_Key_Checks/ Check_Adjustment_Account_ID	Checks for valid values in foreign key column adjustment_account_id in sb_mtl_cc_entries_interface table.	SB-INVCCE-23310
Validate_Foreign_Key_Checks/ Check_Count_Unit_Of_Measure	Checks for valid values in foreign key column count_of UOM in sb_mtl_cc_entries_interface table.	SB-INVCCE-23321
Validate_Foreign_Key_Checks/ Check_Count_UOM	Checks for valid values in foreign key column count_UOM_code in sb_mtl_cc_entries_interface table.	SB-INVCCE-23322
Validate_Foreign_Key_Checks/ Check_Cycle_Count_Entry_ID	Checks for valid values in foreign key column cycle_count_entry_id in sb_mtl_cc_entries_interface table.	SB-INVCCE-23311
Validate_Foreign_Key_Checks/ Check_Cycle_Count_Header_ID	Checks for valid values in foreign key column cycle_count_header_id in sb_mtl_cc_entries_interface table.	SB-INVCCE-23312
Validate_Foreign_Key_Checks/ Check_Inventory_Item_ID	Checks for valid values in foreign key column Inventory_Item_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23319
Validate_Foreign_Key_Checks/ Check_Organization_ID	Checks for valid values in foreign key column Organization_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23316

Table 12 sb_validate_cc_entries_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Locator_ID	Checks for valid values in foreign key column Locator_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23313
Validate_Foreign_Key_Checks/ Check_Lot_Number	Checks for valid values in foreign key column Lot_Number in sb_mtl_cc_entries_interface table.	SB-INVCCE-23315
Validate_Foreign_Key_Checks/ Check_Revision	Checks for valid values in foreign key column Revision in sb_mtl_cc_entries_interface table.	SB-INVCCE-23314
Validate_Foreign_Key_Checks/ Check_Serial_Number	Checks for valid values in foreign key column Serial_Number in sb_mtl_cc_entries_interface table.	SB-INVCCE-23318
Validate_Foreign_Key_Checks/ Check_SubInventory	Checks for valid values in foreign key column Subinventory in sb_mtl_cc_entries_interface table.	SB-INVCCE-23317
Validate_Foreign_Key_Checks/ Check_Transaction_Reason_ID	Checks for valid values in foreign key column Transaction_Reason_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23320
Validate_Required_Columns/ Check_Action_Code	Checks for the required field Action_Code in sb_mtl_cc_entries_interface table.	SB-INVCCE-23303
Validate_Required_Columns/ Check_CC_Entry_Interface_ID	Checks for the required field CC_entry_interface_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23301
Validate_Required_Columns/ Check_Employee_ID	Checks for the required field Employee_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23304
Validate_Required_Columns/ Check_Organization_ID	Checks for the required field Organization_ID in sb_mtl_cc_entries_interface table.	SB-INVCCE-23302

A.1.4 Items

Table 13 sb_validate_items_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Accounting_Period	Derives Acct_period_id from org_acct_periods table.	N/A
Perform_Other_Validations/ Check_Allowed_Units_Code	Checks for valid values in column Allowed_Units_Code of table sb_mtl_system_items_interface.	SB-INVITM-21373
Perform_Other_Validations/ Check_Assembly_Pegging_Flag	Checks for valid values in column Assembly_pegging_flag of table sb_mtl_system_items_interface.	SB-INVITM-21347
Perform_Other_Validations/ Check_ATP_Components_Flag	Checks for valid values in column ATP_components_flag of table sb_mtl_system_items_interface.	SB-INVITM-21331
Perform_Other_Validations/ Check_Auto_Created_Config_Flag	Checks for valid values in column Auto_Created_Config_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21361
Perform_Other_Validations/ Check_BOM_Enabled_Flag	Checks for valid values in column BOM_enabled_flag of table sb_mtl_system_items_interface.	SB-INVITM-21323
Perform_Other_Validations/ Check_BOM_Item_Type	Checks for valid values in column BOM_Item_Type of table sb_mtl_system_items_interface.	SB-INVITM-21372
Perform_Other_Validations/ Check_Buildin_WIP_Flag	Checks for valid values in column Buidin_WIP_flag of table sb_mtl_system_items_interface.	SB-INVITM-21324
Perform_Other_Validations/ Check_Container_Item_Flag	Checks for valid values in column Container_Item_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21362
Perform_Other_Validations/ Check_Costing_Enabled_Flag	Checks for valid values in column Costing_Enabled_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21359
Perform_Other_Validations/ Check_Cust_Order_Flag	Checks for valid values in column customer_order_flag of table sb_mtl_system_items_interface.	SB-INVITM-21310
Perform_Other_Validations/ Check_Cust_Order_Enabled_Flag	Checks for valid values in column cust_order_enabled_flag of table sb_mtl_system_items_interface.	SB-INVITM-21318
Perform_Other_Validations/ Check_Cycle_Count_Enabled_Flag	Checks for valid values in column Cycle_Count_Enabled_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21360
Perform_Other_Validations/ Check_Enabled_Flag	Checks for valid values in column enabled_flag in table sb_mtl_system_items_interface.	SB-INVITM-21307

Table 13 sb_validate_items_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_End_Assembly_Pegging_Flag	Checks for valid values in column End_Assembly_Pegging_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21347
Perform_Other_Validations/ Check_Eng_Item_Flag	Checks for valid values in column Eng_item_flag of table sb_mtl_system_items_interface.	SB-INVITM-21315
Perform_Other_Validations/ Check_Expense_Billable_Flag	Checks for valid values in column Expense_billable_flag of table sb_mtl_system_items_interface.	SB-INVITM-21338
Perform_Other_Validations/ Check_Int_Order_Flag	Checks for valid values in column Internal_order_flag of table sb_mtl_system_items_interface.	SB-INVITM-21311
Perform_Other_Validations/ Check_Int_Order_Enabled_Flag	Checks for valid values in column Internal_order_enabled_flag of table sb_mtl_system_items_interface.	SB-INVITM-21319
Perform_Other_Validations/ Check_Invoice_Enabled_Flag	Checks for valid values in column Invoice_Enabled_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21356
Perform_Other_Validations/ Check_Invoice_Item_Flag	Checks for valid values in column Invoice_Item_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21355
Perform_Other_Validations/ Check_Inv_Asset_Flag	Checks for valid values in column inv_asset_flag of table sb_mtl_system_items_interface.	SB-INVITM-21316
Perform_Other_Validations/ Check_Inv_Item_Flag	Checks for valid values in column Inventory_item_flag of table sb_mtl_system_items_interface.	SB-INVITM-21314
Perform_Other_Validations/ Check_Inv_Planning_Code	Checks for valid values in column Inv_Planning_Code of table sb_mtl_system_items_interface.	SB-INVITM-21374
Perform_Other_Validations/ Check_Item_Desc_Upd_Flag	Checks for valid values in column Item_desc_upd_flag of table sb_mtl_system_items_interface.	SB-INVITM-21325
Perform_Other_Validations/ Check_List_Price_Unit	Checks for valid values in column List_price_unit of table sb_mtl_system_items_interface.	SB-INVITM-21336
Perform_Other_Validations/ Check_Location_Control_Code	Checks for valid values in column Location_Control_Code of table sb_mtl_system_items_interface.	SB-INVITM-21381
Perform_Other_Validations/ Check_Lot_Control_Code	Checks for valid values in column Lot_Control_Code of table sb_mtl_system_items_interface.	SB-INVITM-21366

Table 13 sb_validate_items_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Market_Price	Checks for valid values in column Market_price of table sb_mtl_system_items_interface.	SB-INVITM-21341
Perform_Other_Validations/ Check_Material_Billable_Flag	Checks for valid values in column Material_Billable_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21377
Perform_Other_Validations/ Check_MRP_Planning_Code	Checks for valid values in column MRP_Planning_Code of table sb_mtl_system_items_interface.	SB-INVITM-21380
Perform_Other_Validations/ Check_MRP_Safety_stock_Code	Checks for valid values in column MRP_Safety_stock_Code of table sb_mtl_system_items_interface.	SB-INVITM-21376
Perform_Other_Validations/ Check_MTL_Trans_Enabled_Flag	Checks for valid values in column MTL_trans_enabled_flag of table sb_mtl_system_items_interface.	SB-INVITM-21321
Perform_Other_Validations/ Check_MustUse_ApprVendor_Flag	Checks for valid values in column Must_Use_Approved_Vendor_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21357
Perform_Other_Validations/ Check_Outside_Operation_Flag	Checks for valid values in column Outside_Operation_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21358
Perform_Other_Validations/ Check_Pick_Comps_Flag	Checks for valid values in column Pick_components_flag of table sb_mtl_system_items_interface.	SB-INVITM-21329
Perform_Other_Validations/ Check_Plan_Make_Buy_Code	Checks for valid values in column Plan_Make_Buy_Code of table sb_mtl_system_items_interface.	SB-INVITM-21375
Perform_Other_Validations/ Check_Plan_Time_Fence_Code	Checks for valid values in column Plan_Time_Fence_Code of table sb_mtl_system_items_interface.	SB-INVITM-21371
Perform_Other_Validations/ Check_Plan_Time_Fence_Days	Checks for valid values in column plan_time_fence_days of table sb_mtl_system_items_interface.	SB-INVITM-21328
Perform_Other_Validations/ Check_Post_Proc_Lead_Time	Checks for valid values in column post_proc_lead_time of table sb_mtl_system_items_interface.	SB-INVITM-21348
Perform_Other_Validations/ Check_Prevent_Maint_Flag	Checks for valid values in e column preventive_maintenance_flag of table sb_mtl_system_items_interface.	SB-INVITM-21352
Perform_Other_Validations/ Check_Price_Tolerance_pc	Checks for valid values in column Price_tolerance_percentage of table sb_mtl_system_items_interface.	SB-INVITM-21342

Table 13 sb_validate_items_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Prorate_Service_Flag	ChecksChecks for valid values in column prorate_service_flag of table sb_mtl_system_items_interface.	SB-INVITM-21354
Perform_Other_Validations/ Check_Purchase_Item_Flag	Checks for valid values in column purchase_item_flag of table sb_mtl_system_items_interface.	SB-INVITM-21308
Perform_Other_Validations/ Check_Pur_Enabled_Flag	Checks for valid values in column purchase_enabled_flag of table sb_mtl_system_items_interface.	SB-INVITM-21317
Perform_Other_Validations/ Check_Rate_Decrease	Checks for valid values in column rate_decrease of table sb_mtl_system_items_interface.	SB-INVITM-21345
Perform_Other_Validations/ Check_Rate_Increase	Checks for valid values in column rate_increase of table sb_mtl_system_items_interface.	SB-INVITM-21346
Perform_Other_Validations/ Check_Receipt_Required_Flag	Checks for valid values in column Receipt_required_flag of table sb_mtl_system_items_interface.	SB-INVITM-21326
Perform_Other_Validations/ Check_Rep_Plan_Flag	Checks for valid values in column rep_plan_flag of table sb_mtl_system_items_interface.	SB-INVITM-21344
Perform_Other_Validations/ Check_Repl_To_Order_Flag	Checks for valid values in column Repl_to_order_flag of table sb_mtl_system_items_interface.	SB-INVITM-21330
Perform_Other_Validations/ Check_Reservable_Type	Checks for valid values in column Reservable_Type of table sb_mtl_system_items_interface.	SB-INVITM-21378
Perform_Other_Validations/ Check_Restrict_Locators_Code	Checks for valid values in column Restrict_Locators_Code of table sb_mtl_system_items_interface.	SB-INVITM-21370
Perform_Other_Validations/ Check_Return_Insp_Reqmt	Checks for valid values in column Return_Insp_Reqmt of table sb_mtl_system_items_interface.	SB-INVITM-21379
Perform_Other_Validations/ Check_RFQ_Reqd_Flag	Checks for valid values in column RFQ_required_flag of table sb_mtl_system_items_interface.	SB-INVITM-21327
Perform_Other_Validations/ Check_Service_Duration	Checks for valid values in column Service_duration of table sb_mtl_system_items_interface.	SB-INVITM-21340
Perform_Other_Validations/ Check_Service_Item_Flag	Checks for valid values in column Service_item_flag of table sb_mtl_system_items_interface.	SB-INVITM-21312

Table 13 sb_validate_items_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Service_Start_Delay	Checks for valid values in column service_start_delay of table sb_mtl_system_items_interface.	SB-INVITM-21313
Perform_Other_Validations/ Check_Serv_Comp_Flag	Checks for valid values in column Serv_comp_flag of table sb_mtl_system_items_interface.	SB-INVITM-21350
Perform_Other_Validations/ Check_Serv_Product_Flag	Checks for valid values in column Servi_product_flag duration of table sb_mtl_system_items_interface.	SB-INVITM-21351
Perform_Other_Validations/ Check_Shelf_Life	Checks for valid values in column shelf_life of table sb_mtl_system_items_interface.	SB-INVITM-21343
Perform_Other_Validations/ Check_Shelf_Life_Code	Checks for valid values in column Shelf_Life_Code of table sb_mtl_system_items_interface.	SB-INVITM-21367
Perform_Other_Validations/ Check_Ship_Item_Flag	Checks for valid values in column ship_item_flag of table sb_mtl_system_items_interface.	SB-INVITM-21309
Perform_Other_Validations/ Check_Ship_Model_Complete_Flag	Checks for valid values in column Ship_Model_Complete_flag of table sb_mtl_system_items_interface.	SB-INVITM-21353
Perform_Other_Validations/ Check_SI_No_Control_Code	Checks for valid values in column SI_No_Control_Code of table sb_mtl_system_items_interface.	SB-INVITM-21368
Perform_Other_Validations/ Check_SO_Trans_Flag	Checks for valid values in column SO_Trans_flag of table sb_mtl_system_items_interface.	SB-INVITM-21320
Perform_Other_Validations/ Check_Stock_Enabled_Flag	Checks for valid values in column Stock_Enabled_flag of table sb_mtl_system_items_interface.	SB-INVITM-21322
Perform_Other_Validations/ Check_Sub_Inv_Code	Checks for valid values in column Sub_Inv_Code of table sb_mtl_system_items_interface.	SB-INVITM-21369
Perform_Other_Validations/ Check_Summary_Flag	Checks for valid values in column Summary_flag in table sb_mtl_system_items_interface.	SB-INVITM-21306
Perform_Other_Validations/ Check_Taxable_Flag	Checks for valid values in column Taxable_flag of table sb_mtl_system_items_interface.	SB-INVITM-21337
Perform_Other_Validations/ Check_Time_Billable_Flag	Checks for valid values in column Time_Billable_flag of table sb_mtl_system_items_interface.	SB-INVITM-21339

Table 13 sb_validate_items_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Vehicle_Item_Flag	Checks for valid values in column Vehicle_Item_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21363
Perform_Other_Validations/ Check_Vendor_Warranty_Flag	Checks for valid values in column Vendor_Warranty_Flag of table sb_mtl_system_items_interface.	SB-INVITM-21349
Perform_Other_Validations/ Check_WIP_Supply_Type	Checks for valid values in column WIP_Supply_Type of table sb_mtl_system_items_interface.	SB-INVITM-21365
Validate_Foreign_Key_Checks/ Check_Cost_of_Sales_Act	Checks for valid values in column cost_of_sales_account of table sb_mtl_system_items_interface.	SB-INVITM-21332
Validate_Foreign_Key_Checks/ Check_Encumbrance_Account	Checks for valid values in column Encumbrance_account of table sb_mtl_system_items_interface.	SB-INVITM-21334
Validate_Foreign_Key_Checks/ Check_Expense_Account	Checks for valid values in column Expense_account of table sb_mtl_system_items_interface.	SB-INVITM-21335
Validate_Foreign_Key_Checks/ Check_Org_ID_Code	Checks for valid values in foreign key column organization_id in table sb_mtl_system_items_interface.	SB-INVITM-21364
Validate_Foreign_Key_Checks/ Check_Sales_Act_DSP	Checks for valid values in column sales_account of table sb_mtl_system_items_interface.	SB-INVITM-21333
Validate_Required_Columns/ Check_Item	Checks for the required field item_number in table sb_mtl_system_items_interface.	SB-INVITM-21302
Validate_Required_Columns/ Check_Item_Desc	Checks for the required field Item_Desc in table sb_mtl_system_items_interface.	SB-INVITM-21382
Validate_Required_Columns/ Check_Organization_Id	Checks for the required field organization_id in table sb_mtl_system_items_interface.	SB-INVITM-21301
Validate_Required_Columns/ Check_Process_Flag	Checks for the required field process_flag in table sb_mtl_system_items_interface.	SB-INVITM-21303
Validate_Required_Columns/ Check_Proc_Id	Checks for the required field set_process_id in table sb_mtl_system_items_interface.	SB-INVITM-21305
Validate_Required_Columns/ Check_Tran_Type	Checks for the required field Transaction_Type in table sb_mtl_system_items_interface.	SB-INVITM-21304

A.1.5 Item Revisions

Table 14 sb_validate_invrev_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Item_Number	Checks for valid values in column Item_Number in sb_mtl_item_revs_interface table.	SB-INVREV-22509
Perform_Other_Validations/ Check_Process_flag	Checks for valid values in column Process_flag in sb_mtl_item_revs_interface table.	SB-INVREV-22508
Validate_Foreign_Key_Checks/ Check_Change_Notice	Checks for valid values in foreign key column Change_Notice in sb_mtl_item_revs_interface table.	SB-INVREV-22504
Validate_Foreign_Key_Checks/ Check_Inventory_item_id	Checks for valid values in foreign key column Inventory_item_id in sb_mtl_item_revs_interface table.	SB-INVREV-22506
Validate_Foreign_Key_Checks/ Check_Organization_id	Checks for valid values in foreign key column Organization_id in sb_mtl_item_revs_interface table.	SB-INVREV-22505
Validate_Foreign_Key_Checks/ Check_Primary_key	Checks for valid values in foreign key column Primary_key in sb_mtl_item_revs_interface table.	SB-INVREV-22507
Validate_Required_Columns/ Check_Inventory_Item_Id	Checks for the field Inventory_Item_Id in sb_mtl_item_revs_interface table.	SB-INVREV-22501
Validate_Required_Columns/ Check_Organization_Id	Checks for the field Organization_Id in sb_mtl_item_revs_interface table.	SB-INVREV-22502
Validate_Required_Columns/ Check_Revision	Checks for the field Revision in sb_mtl_item_revs_interface table.	SB-INVREV-22503
Validate_Required_Columns/ Check_Set_process_ID	Checks for the field Set_process_ID in sb_mtl_item_revs_interface table.	SB-INVREV-22500

A.1.6 Replenishment

Master

Table 15 sb_validate_rep_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_REC/Validate	This is the main validation procedure of Replenishment Interface.	SB-INVREP-22430

Headers

Table 16 sb_validate_rep_hdrs_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Error_Flag	Checks for Error_Flag field in table mtl_replenish_headers_int.	SB-INVREP-22314
Perform_Other_Validations/ Check_Last_Update	Checks that value in field Last_updated_by is a valid user_id in fnd_user in table mtl_replenish_headers_int.	SB-INVREP-22319
Perform_Other_Validations/ Check_Lines	Checks that there is at least one line per Header in table mtl_replenish_headers_int.	SB-INVREP-22320
Perform_Other_Validations/ Check_Process_Mode	Checks that the value in field Process_Modes is 1, 2, or 3 in table mtl_replenish_headers_int.	SB-INVREP-22315
Perform_Other_Validations/ Check_Process_Status	Checks that the value in field Process_Status is 1, 2, 3, 4, or 5 in table mtl_replenish_headers_int.	SB-INVREP-22316
Perform_Other_Validations/ Check_Replenish_Header	Checks that the value in field Replenishment_header_id is unique for the base table in table mtl_replenish_headers_int.	SB-INVREP-22317
Perform_Other_Validations/ Check_Replenish_Count	Checks that the value in field Replenishment_Count_Name is unique for the base table for existing count headers in table mtl_replenish_headers_int.	SB-INVREP-22318
Validate_Foreign_Key_Checks/ Check_Organization_ID	Checks validity of value for Organization_ID in MTL_Parameters table in table mtl_replenish_headers_int.	SB-INVREP-22312

Table 16 sb_validate_rep_hdrs_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_SubInventory_Code	Checks validity of values for Organization_ID and SubInventory_Code in MTL_Secondary_Inventories table.	SB-INVREP-22313
Validate_Required_Columns/ Check_Count_Date	Checks for non-null entry in field Count_Date in table mtl_replenish_headers_int.	SB-INVREP-22303
Validate_Required_Columns/ Check_Created_By	Checks for non-null entry in field Check_Created_By in table mtl_replenish_headers_int.	SB-INVREP-22306
Validate_Required_Columns/ Check_Creation_Date	Checks for non-null entry in field Creation_Date in table mtl_replenish_headers_int.	SB-INVREP-22305
Validate_Required_Columns/ Check_Last_Update_Date	Checks for non-null entry in field Last_Update_Date in table mtl_replenish_headers_int.	SB-INVREP-22304
Validate_Required_Columns/ Check_Last_Updated_By	Checks for non-null entry in field Last_Updated_By in table mtl_replenish_headers_int.	SB-INVREP-22307
Validate_Required_Columns/ Check_Organization_ID	Checks for non-null entry in field Organization_ID in table mtl_replenish_headers_int.	SB-INVREP-22308
Validate_Required_Columns/ Check_Process_Mode	Checks for non-null entry in field Process_Mode in table mtl_replenish_headers_int.	SB-INVREP-22311
Validate_Required_Columns/ Check_Process_Status	Checks for non-null entry in field SubInventory_Code in table mtl_replenish_headers_int.	SB-INVREP-22310
Validate_Required_Columns/ Check_Replenishment_Count_Name	Checks for non-null entry in field Replenishment_Count_Name in table mtl_replenish_headers_int.	SB-INVREP-22302
Validate_Required_Columns/ Check_Replenishment_Header_ID	Checks for non-null entry in field Replenishment_Header_ID in table mtl_replenish_headers_int.	SB-INVREP-22301
Validate_Required_Columns/ Check_SubInventory_Code	Checks for non-null entry in field Last_Updated_By in table mtl_replenish_headers_int.	SB-INVREP-22309

Lines

Table 17 sb_validate_rep_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Count_Type_Code	Checks that the value in field Count_Type_Code is 1, 2, or 3 in table mtl_replenish_lines_int.	SB-INVREP-22419
Perform_Other_Validations/ Check_Count_Type_Tracked	Checks that value in Count_Type_Code field is consistent with sub inventory being tracked in table mtl_replenish_lines_int.	SB-INVREP-22421
Perform_Other_Validations/ Check_Count_Type_Tracked23	Checks that value in Count_Type_Code field is consistent with sub inventory being non_tracked in table mtl_replenish_lines_int.	SB-INVREP-22421
Perform_Other_Validations/ Check_Count_uom_primary	Checks that a conversion to primary UOM exists for the value in Count_uom_primary field in table mtl_replenish_lines_int.	SB-INVREP-22424
Perform_Other_Validations/ Check_Inventory_item	Checks that value in Inventory_item field refers to a transactable item in the organization specified in table mtl_replenish_lines_int.	SB-INVREP-22422
Perform_Other_Validations/ Check_MinMAX_Plan	Checks that value in MinMAX_Plan field is a Min-Max planned for that sub inventory in table mtl_replenish_lines_int.	SB-INVREP-22423
Perform_Other_Validations/ Check_Organization_Id	Checks that Organization_Id is the same as in the Header level in table mtl_replenish_lines_int.	SB-INVREP-22420
Validate_Foreign_Key_Checks/ Check_Count_Uom	Checks that Count_Unit_of_Measure is a valid Unit_of_measure in table mtl_replenish_lines_int.	SB-INVREP-22416
Validate_Foreign_Key_Checks/ Check_Count_Uom_Code	Checks that Count_Uom_Code is a valid Uom_Code in table mtl_replenish_lines_int.	SB-INVREP-22417
Validate_Foreign_Key_Checks/ Check_Inventory_Item_ID	Checks validity of values for Inventory_Item_ID and Organization_ID in table mtl_replenish_lines_int.	SB-INVREP-22415
Validate_Foreign_Key_Checks/ Check_Organization_ID	Checks validity of value for Organization_ID in table mtl_replenish_lines_int.	SB-INVREP-22413

Table 17 sb_validate_rep_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Rep_Header_ID	Checks validity of value for Organization_ID in table mtl_replenish_lines_int.	SB-INVREP-22414
Validate_Foreign_Key_Checks/ Check_Segment	Checks validity of values for Segments 1 - 20 in table mtl_replenish_lines_int.	SB-INVREP-22418
Validate_Required_Columns/ Check_Count_Quantity	Checks for non-null entry in field Count_Quantity in table mtl_replenish_lines_int.	SB-INVREP-22410
Validate_Required_Columns/ Check_Count_Type_Code	Checks for non-null entry in field Count_Type_Code in table mtl_replenish_lines_int.	SB-INVREP-22409
Validate_Required_Columns/ Check_Count_UOM	Checks for non-null entry in field Count_Unit_Of_Measure in table mtl_replenish_lines_int.	SB-INVREP-22411
Validate_Required_Columns/ Check_Count_UOM_Code	Checks for non-null entry in field Count_UOM_Code in table mtl_replenish_lines_int.	SB-INVREP-22412
Validate_Required_Columns/ Check_Created_By	Checks for non-null entry in field Created_By in table mtl_replenish_lines_int.	SB-INVREP-22405
Validate_Required_Columns/ Check_Creation_Date	Checks for non-null entry in field Creation_Date in table mtl_replenish_lines_int.	SB-INVREP-22404
Validate_Required_Columns/ Check_Inventory_Item_ID	Checks for non-null entry in field Inventory_Item_ID in table mtl_replenish_lines_int.	SB-INVREP-22408
Validate_Required_Columns/ Check_Last_Update_Date	Checks for non-null entry in field Last_Update_Date in table mtl_replenish_lines_int.	SB-INVREP-22403
Validate_Required_Columns/ Check_Last_Update_Login	Checks for non-null entry in field Last_Update_Login in table mtl_replenish_lines_int.	SB-INVREP-22406
Validate_Required_Columns/ Check_Last_Updated_By	Checks for non-null entry in field Last_Updated_By in table mtl_replenish_lines_int.	SB-INVREP-22407
Validate_Required_Columns/ Check_Replenishment_Header_ID	Checks for non-null entry in field Replenishment_Header_ID in table mtl_replenish_lines_int.	SB-INVREP-22401
Validate_Required_Columns/ Check_Replenishment_Line_Id	Checks for non-null entry in field Replenishment_Line_Id in table mtl_replenish_lines_int.	SB-INVREP-22402

A.1.7 Transactions

Table 18 sb_validate_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Derive_Acct_Period_id	Derives appropriate values for the column acct_period_id in sb_mtl_transactions_interface.	N/A
Derive_Required_Columns/ Derive_primary_quantity	Derives appropriate values for the column primary_quantity in sb_mtl_transactions_interface.	N/A
Derive_Required_Columns/ Derive_tran_action_id	Derives appropriate values for the column transaction_action_id in sb_mtl_transactions_interface.	N/A
Derive_Required_Columns/ Derive_tran_source_type_id	Derives appropriate values for the column transaction_source_type_id in table sb_mtl_transactions_interface.	N/A
Perform_Other_Validations/ Check_WIP_Entity_Type	Checks for appropriate value in the column wip_entity_type in table sb_mtl_transactions_interface.	SB-INVTXN-20334
Perform_Other_Validations/ Update_Fixed_Values	This routine updates some columns with their fixed values in table sb_mtl_transactions_interface.	N/A
Perform_Other_Validations/ Check_Reason_ID	Checks for appropriate value in the column reason_id in table sb_mtl_transactions_interface.	SB-INVTXN-20333
Perform_Other_Validations/ Check_Substitution_ID	Checks for appropriate value in the substitution_id column of table sb_mtl_transactions_interface.	SB-INVTXN-20335
Perform_Other_Validations/ Check_Trans_Header_ID	Checks for the appropriate value in the transaction_header_id column of table sb_mtl_transactions_interface.	SB-INVTXN-20344
Validate_Foreign_Key_Checks/ Check_Acct_Period_ID	Checks for valid values in foreign key column acct_period_id in table sb_mtl_transactions_interface.	SB-INVTXN-20322
Validate_Foreign_Key_Checks/ Check_Distribution_Account_ID	Checks for valid values in foreign key column distribution_account_id in table sb_mtl_transactions_interface.	SB-INVTXN-20319
Validate_Foreign_Key_Checks/ Check_Inventory_Item_ID	Checks for valid values in foreign key column inventory_item_id in table sb_mtl_transactions_interface.	SB-INVTXN-20320
Validate_Foreign_Key_Checks/ Check_Locator_ID	Checks for valid values in foreign key column locator_id in table sb_mtl_transactions_interface.	SB-INVTXN-20326

Table 18 sb_validate_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Operation_Seq_Num	Checks for valid values in foreign key column operation_seq_num in table sb_mtl_transactions_interface.	SB-INVTXN-20324
Validate_Foreign_Key_Checks/ Check_Organization_ID	Checks for valid values in foreign key column organization_id in table sb_mtl_transactions_interface.	SB-INVTXN-20321
Validate_Foreign_Key_Checks/ Check_Revision	Checks for valid values in foreign key column revision in table sb_mtl_transactions_interface.	SB-INVTXN-20329
Validate_Foreign_Key_Checks/ Check_Schedule_ID	Checks for valid values in foreign key column schedule_id in table sb_mtl_transactions_interface.	SB-INVTXN-20323
Validate_Foreign_Key_Checks/ Check_Schedule_Number	Checks for valid values in foreign key column schedule_number in table sb_mtl_transactions_interface.	SB-INVTXN-20332
Validate_Foreign_Key_Checks/ Check_Sub_inventory_Code	Checks for valid values in foreign key column sub_inventory_code in table sb_mtl_transactions_interface.	SB-INVTXN-20330
Validate_Foreign_Key_Checks/ Check_Transfer_Locator	Checks for valid values in foreign key column transfer_locator in table sb_mtl_transactions_interface.	SB-INVTXN-20328
Validate_Foreign_Key_Checks/ Check_Transfer_Organization	Checks for valid values in foreign key column transfer_organization in table sb_mtl_transactions_interface.	SB-INVTXN-20327
Validate_Foreign_Key_Checks/ Check_Transfer_SubInventory	Checks for valid values in foreign key column transfer_subinventory in table sb_mtl_transactions_interface.	SB-INVTXN-20331
Validate_Foreign_Key_Checks/ Check_Transportation_Account	Checks for valid values in foreign key column transportation_account in table sb_mtl_transactions_interface.	SB-INVTXN-20325
Validate_Required_Columns/ Check_Inventory_Item_ID	Checks for the required field inventory_item_id in table sb_mtl_transactions_interface.	SB-INVTXN-20307
Validate_Required_Columns/ Check_Organization_ID	Checks for the required field organization_id in table sb_mtl_transactions_interface.	SB-INVTXN-20308
Validate_Required_Columns/ Check_Process_Flag	Checks for the required field process_flag in table sb_mtl_transactions_interface.	SB-INVTXN-20304
Validate_Required_Columns/ Check_Source_Code;	Checks for the required field source_code in table sb_mtl_transactions_interface.	SB-INVTXN-20301

Table 18 sb_validate_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Required_Columns/ Check_Source_Header_ID;	Checks for the required field source_header_id in table sb_mtl_transactions_interface.	SB-INVTXN-20303
Validate_Required_Columns/ Check_Source_Line_ID;	Checks for the required field source_line_id in table sb_mtl_transactions_interface.	SB-INVTXN-20302
Validate_Required_Columns/ Check_Transaction_Date	Checks for the required field transaction_date in table sb_mtl_transactions_interface.	SB-INVTXN-20311
Validate_Required_Columns/ Check_Transaction_Interface_ID	Checks for the required field transaction_interface_id in table sb_mtl_transactions_interface.	SB-INVTXN-20306
Validate_Required_Columns/ Check_Transaction_Mode	Checks for the required field transaction_mode in table sb_mtl_transactions_interface.	SB-INVTXN-20305
Validate_Required_Columns/ Check_Transaction_Quantity	Checks for the required field transaction_quantity in table sb_mtl_transactions_interface.	SB-INVTXN-20309
Validate_Required_Columns/ Check_Transaction_Source_ID	Checks for the required field transaction_source_id in table sb_mtl_transactions_interface.	SB-INVTXN-20312
Validate_Required_Columns/ Check_Transaction_Source_Name	Checks for the required field transaction_source_name in table sb_mtl_transactions_interface.	SB-INVTXN-20313
Validate_Required_Columns/ Check_Transaction_Type_ID	Checks for the required field transaction_type_id in table sb_mtl_transactions_interface.	SB-INVTXN-20314
Validate_Required_Columns/ Check_Transaction_UOM	Checks for the required field transaction_UOM in table sb_mtl_transactions_interface.	SB-INVTXN-20310

Lots

Table 19 sb_validate_trans_lots_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Derive_primary_quantity	Derives appropriate values for the column primary_quantity in sb_mtl_trans_lots_interface.	N/A
Validate	This procedure performs all pre-validations pertaining to the Transactions Interface Table.	N/A
Validate_Foreign_Key_Checks/ Check_Serial_Tran_Temp_ID	Checks for valid values in foreign key column serial_transaction_temp_id in table sb_mtl_trans_lots_interface.	SB-INVTXN-20339
Validate_Foreign_Key_Checks/ Check_Tran_Interface_ID	Checks for valid values in foreign key column Transaction_interface_id in table sb_mtl_trans_lots_interface.	SB-INVTXN-20340
Validate_Required_Columns/ Check_Lot_Number	Checks for the required field lot_number in table sb_mtl_trans_lots_interface.	SB-INVTXN-20337
Validate_Required_Columns/ Check_Tran_Interface_ID	Checks for the required field Transaction_interface_id in table sb_mtl_trans_lots_interface.	SB-INVTXN-20336
Validate_Required_Columns/ Check_Tran_Quantity	Checks for the required field transaction_quantity in table sb_mtl_trans_lots_interface.	SB-INVTXN-20338

Serial Numbers

Table 20 sb_validate_serial_nos_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Vendor_Serial_No	Checks for valid values in foreign key column vendor_serial_number in table sb_mtl_serial_nos_interface.	SB-INVTXN-20343
Validate_Required_Columns/ Check_Fm_Serial_Number	Checks for the required field fm_serial_number in table sb_mtl_serial_nos_interface.	SB-INVTXN-20342
Validate_Required_Columns/ Check_Tran_Interface_ID	Checks for the required field transaction_interface_id in table sb_mtl_serial_nos_interface.	SB-INVTXN-20341

A.2 Purchase Order

A.2.1 Import

Master

Table 21 sb_validate_poi_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate	Performs all pre-validations pertaining to Purchase Order Import.	SB-POIMP-25006
Validate_Master_Details	Performs all validations that require checking both the master and detail records across interface tables of Purchase Order Import.	SB-POIMP-20001

Headers

Table 22 sb_validate_poi_hdrs_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Check_For_Errors	Checks if an error has occurred by checking the count of updated records for errors.	None
Derive_Required_Columns/ Derive_Po_Header_ID	Po_Header_Id is derived from the Document Type and Document Number.	N/A
Derive_Required_Columns/ Derive_Vendor_ID	Vendor_Id is derived from the given Vendor Name.	N/A
Derive_Required_Columns/ Derive_Vendor_Site_ID	Vendor_Site_Id is derived from the Vendor Id and Vendor Site Code.	N/A
Perform_Other_Validations/ Check_Action_For_Std_Doc_Type	Checks if Document Type Code is STANDARD and Action is neither UPDATE nor REPLACE.	SB-POIMP-25007
Perform_Other_Validations/ Check_Amount_Agreed	Checks if Amount_Agreed is more than Amount_Limit.	SB-POIMP-20031
Perform_Other_Validations/ Check_Amount_Limit	Checks if Amount_limit is more than Zero.	SB-POIMP-20033
Perform_Other_Validations/ Chk_App_Stat_For_Std_Doc_Type	Checks if Document Type is STANDARD and Approval Status is INCOMPLETE.	SB-POIMP-25008
Perform_Other_Validations/ Check_Document_Sub_Type	Checks for appropriate value in the Document_Sub_Type column of table sb_po_headers_interface.	SB-POIMP-25016

Table 22 sb_validate_poi_hdrs_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Lines_For_Header	Checks if there exists at least one Line record per Header record in the SB_PO_LINES_INTERFACE table.	SB-POIMP-25013
Perform_Other_Validations/ Check_Min_Release_Amount	Checks if Min_Release_Amount is more than Amount_Limit.	SB-POIMP-20032
Perform_Other_Validations/ Check_Num_Of_Lines	Checks for appropriate value in the Num_Of_Lines column of table sb_po_headers_interface.	SB-POIMP-25017
Perform_Other_Validations/ Check_Release_Date	Checks if Release_Date is NULL since it is reserved for future use.	SB-POIMP-25011
Perform_Other_Validations/ Check_Release_Num	Checks if Release_Num is NULL since it is reserved for future use.	SB-POIMP-20088
Perform_Other_Validations/ Check_Vendor_Doc_Num_Org_Blkt	Checks if Action is ORIGINAL and Document Type Code is BLANKET and Vendor Doc Num does not exist in the PO_HEADERS_ALL table.	SB-POIMP-25010
Perform_Other_Validations/ Check_Vendor_Doc_Num_Org_Quot	Checks if Action is ORIGINAL and Document Type Code is QUOTATION and Vendor Doc Num does not exist in the PO_HEADERS_ALL table.	SB-POIMP-25010
Perform_Other_Validations/ Check_Vendor_Doc_Num_Repl_Blkt	Checks if Action is REPLACE and Document Type Code is BLANKET and Vendor Doc Num exists in the PO_HEADERS_ALL table.	SB-POIMP-25009
Perform_Other_Validations/ Check_Vendor_Doc_Num_Repl_Quot	Checks if Action is REPLACE and Document Type Code is QUOTATION and Vendor Doc Num exists in the PO_HEADERS_ALL table.	SB-POIMP-25009
Perform_Other_Validations/ Check_Vendor_Doc_Num_Upd_Blkt	Checks if Action is UPDATE and Document Type Code is BLANKET and Vendor Doc Num exists in the PO_HEADERS_ALL table.	SB-POIMP-25009
Perform_Other_Validations/ Check_Vendor_Doc_Num_Upd_Quot	Checks if Action is UPDATE and Document Type Code is QUOTATION and Vendor Doc Num exists in the PO_HEADERS_ALL table.	SB-POIMP-25009
Validate_Foreign_Key_Checks/ Check_Agent_ID	Checks if Agent_Id is a valid value in the PO_AGENTS table.	SB-POIMP-20015
Validate_Foreign_Key_Checks/ Check_Agent_Name	Checks if Agent Name is valid.	SB-POIMP-20023
Validate_Foreign_Key_Checks/ Check_Bill_To_Location_Code	Checks if Bill_To_Location_Code is valid.	SB-POIMP-20028

Table 22 sb_validate_poi_hdrs_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Bill_To_Location_ID	Checks if Bill_To_Location_Id is a valid value in the HR_LOCATIONS_ALL table.	SB-POIMP-20016
Validate_Foreign_Key_Checks/ Check_Currency_Code	Checks if Currency_Code is a valid value in the FND_CURRENCIES table.	SB-POIMP-20021
Validate_Foreign_Key_Checks/ Check_Document_Num	Checks if Document Number is a valid value in the PO_HEADERS_ALL table.	SB-POIMP-20022
Validate_Foreign_Key_Checks/ Check_Document_Num_For_Orig	Checks if the Document Num already exists for an ORIGINAL Purchase Order.	SB-POIMP-25012
Validate_Foreign_Key_Checks/ Check_Fob_Active	Checks whether field Fob_Active is Active or Inactive.	SB-POIMP-25014
Validate_Foreign_Key_Checks/ Check_Freight_Terms_Active	Checks whether field Freight_Terms_Active is Active or Inactive.	SB-POIMP-25015
Validate_Foreign_Key_Checks/ Check_From_Header_ID	Checks if From_Header_Id is a valid value in the PO_HEADERS_ALL table	SB-POIMP-20019
Validate_Foreign_Key_Checks/ Check_From_Type_Lookup_Code	Checks if From_Type_Lookup_Code is a valid value in the PO_DOCUMENT_TYPES_ALL_B table.	SB-POIMP-20011
Validate_Foreign_Key_Checks/ Check_Payment_Terms	Checks if Payment_Terms is valid.	SB-POIMP-20029
Validate_Foreign_Key_Checks/ Check_Rate_Type	Checks if Rate_Type is a valid value in the GL_DAILY_CONVERSION_TYPES table.	SB-POIMP-20020
Validate_Foreign_Key_Checks/ Check_Rfq_Num	Checks if Rfq_Num is valid.	SB-POIMP-20030
Validate_Foreign_Key_Checks/ Check_Ship_To_Location_Code	Checks if Ship_To_Location_Code is valid.	SB-POIMP-20027
Validate_Foreign_Key_Checks/ Check_Ship_To_Location_ID	Checks if Ship_To_Location_Id is a valid value in the HR_LOCATIONS_ALL table.	SB-POIMP-20012
Validate_Foreign_Key_Checks/ Check_Terms_ID	Checks if Terms_Id is a valid value in the AP_TERMS_TL table.	SB-POIMP-20013
Validate_Foreign_Key_Checks/ Check_Ussgl_Transaction_Code	Checks Ussgl_Transaction_Code is a valid value in the GL_USSGL_TRANSACTION_CODES table.	SB-POIMP-20014
Validate_Foreign_Key_Checks/ Check_Vendor_Contact	Checks if Vendor Contact is valid.	SB-POIMP-20026

Table 22 sb_validate_poi_hdrs_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Vendor_Contact_ID	Checks if Vendor_Contact_Id is a valid value in the PO_VENDOR_CONTACTS table.	SB-POIMP-20018
Validate_Foreign_Key_Checks/ Check_Vendor_ID	Checks if Vendor ID supplied is a valid value in the PO_VENDORS table.	SB-POIMP-20010
Validate_Foreign_Key_Checks/ Check_Vendor_Name	Checks if Vendor Name or Vendor Num is valid.	SB-POIMP-20024
Validate_Foreign_Key_Checks/ Check_Vendor_Site_Code	Checks validity of Vendor Site Code.	SB-POIMP-20025
Validate_Foreign_Key_Checks/ Check_Vendor_Site_ID	Checks if Vendor_Site_Id is a valid value in the PO_VENDOR_SITES_ALL table.	SB-POIMP-20017
Validate_Required_Columns/ Check_Action	Checks the Action column. It should be entered since it is a mandatory Column.	SB-POIMP-20003
Validate_Required_Columns/ Check_Action_Values	Checks if Action field has a value other than one of ADD, NEW, ORIGINAL, REPLACE or UPDATE.	SB-POIMP-20008
Validate_Required_Columns/ Check_Doc_Type_Code_Values	Checks if Document Type Code field has a value other than one of PO, RFQ, BLANKET, STANDARD or QUOTATION.	SB-POIMP-20009
Validate_Required_Columns/ Check_Document_Type_Code	Checks the Document Type Code column. It should be entered since it is a mandatory Column.	SB-POIMP-20004
Validate_Required_Columns/ Check_Interface_Header_ID	Checks the Interface Header ID column. It should be entered since it is a mandatory Column.	SB-POIMP-20002
Validate_Required_Columns/ Check_Vendor_Doc_Num	Checks the Vendor Doc Num column. It should be entered since it is a mandatory Column.	SB-POIMP-20007
Validate_Required_Columns/ Check_Vendor_Site_Code	Checks existence of (mandatory) Vendor Site Code column.	SB-POIMP-20006

Lines

Table 23 sb_validate_poi_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Check_For_Errors	Checks if an error has occurred by checking the count of updated records for errors.	None
Derive_Required_Columns/ Derive_Po_Header_ID	Po_Header_Id is derived from the corresponding header's Document Type and Document Num.	N/A
Perform_Other_Validations/ Check_Cap_Exp_Flag_For_Quot	Checks if Capital_Expense_Flag is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20075
Perform_Other_Validations/ Check_Capital_Exp_Flg	Checks if Capital_Expense_Flag is 'Y' or 'N' if Header Document Type is 'BLANKET' or 'STANDARD'.	SB-POIMP-25004
Perform_Other_Validations/ Check_Closed_By	Checks if Closed_By is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20098
Perform_Other_Validations/ Check_Closed_Code_For_Quot	Checks if Closed_Code is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20069
Perform_Other_Validations/ Check_Closed_Date	Checks if Closed_Date is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20097
Perform_Other_Validations/ Check_Closed_Reason	Checks if Closed_Reason is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20096
Perform_Other_Validations/ Check_Cmtd_Amt_For_Quot	Checks if Committed_Amount is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20070
Perform_Other_Validations/ Check_Committed_Amount	Checks if Committed_Amount is greater than 0 if Header Document Type is 'BLANKET' or 'STANDARD'.	SB-POIMP-25001
Perform_Other_Validations/ Check_Contract_Num	Checks if Contract_Num is NULL since it is reserved for future use.	SB-POIMP-20091
Perform_Other_Validations/ Check_Firm_Flag	Checks if Firm_Flag is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20099
Perform_Other_Validations/ Check_Fob_For_Blkt	Checks if Fob is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20083
Perform_Other_Validations/ Check_Freight_Terms_For_Blkt	Checks if Freight_Terms is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20084

Table 23 sb_validate_poi_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Insp_Req_Flag_For_Blkt	Checks if Inspection_Required_Flag is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20086
Perform_Other_Validations/ Check_Market_Price	Checks if Market_Price is greater than 0 if Header Document Type is 'BLANKET' or 'STANDARD'.	SB-POIMP-25002
Perform_Other_Validations/ Check_Market_Price_For_Quot	Checks if Market_Price is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20071
Perform_Other_Validations/ Check_Max_Ord_Qty_For_Quot	Checks if Max_Order_Quantity is greater than 0 and NOT NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20078
Perform_Other_Validations/ Check_Min_Ord_Qty_For_Quot	Checks if Min_Order_Quantity is greater than 0 and NOT NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20077
Perform_Other_Validations/ Check_Min_Rel_Amt	Checks if Min_Release_Amt is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20076
Perform_Other_Validations/ Check_Need_By_Date	Checks if Need_By_Date is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20094
Perform_Other_Validations/ Check_Neg_By_Prep_Flg	Checks if Negotiated_By_Preparer_Flag is 'Y' or 'N' if Header Document Type is 'BLANKET' or 'STANDARD'.	SB-POIMP-25003
Perform_Other_Validations/ Check_Not_To_Exceed_Price	Checks if Not_To_Exceed_Price is NULL if Allow_Price_Override_Flag is 'N' for Document Types STANDARD, BLANKET.	SB-POIMP-20066
Perform_Other_Validations/ Check_Not_To_Exc_Prc_For_Quot	Checks if Not_To_Exceed_Price is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20073
Perform_Other_Validations/ Check_Order_Quantity	Checks if Max_Order_Quantity is greater than Min_Order_Quantity.	SB-POIMP-20068
Perform_Other_Validations/ Check_Ovr_Tol_Error_Flg	Checks if Over_Tolerance_Error_Flag is 'Y' or 'N' if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20079
Perform_Other_Validations/ Check_Payment_Terms	Checks if Payment Terms is valid.	SB-POIMP-20063
Perform_Other_Validations/ Check_Po_Header_ID	Checks if Po_Header_Id is a valid value in the PO_HEADERS_ALL table.	SB-POIMP-20037
Perform_Other_Validations/ Check_Po_Release_Id	Checks if Po_Release_Id is NULL since it is reserved for future use.	SB-POIMP-20089

Table 23 sb_validate_poi_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Promised_Date	Checks if Promised_Date is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20095
Perform_Other_Validations/ Check_Quantity	Checks if Quantity is greater than Zero and not NULL.	SB-POIMP-20100
Perform_Other_Validations/ Check_Rcpt_Days_Exp_Cd	Checks if Receipt_Days_Exception_Code is NULL if Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-20093
Perform_Other_Validations/ Check_Rcpt_Req_Flag_For_Blkt	Checks if Receipt_Required_Flag is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20085
Perform_Other_Validations/ Check_Release_Num	Checks if Release_Num is NULL since it is reserved for future use.	SB-POIMP-20088
Perform_Other_Validations/ Check_Source_Shipment_Id	Checks if Source_Shipment_Id is NULL since it is reserved for future use.	SB-POIMP-20090
Perform_Other_Validations/ Check_Taxable_Flag	Checks if Tax_Name or Tax_Code_Id is NULL if Taxable_Flag is NULL or N and 'N' and Header Document Type is 'BLANKET' or 'QUOTATION'.	SB-POIMP-25005
Perform_Other_Validations/ Check_Type_1099	Checks if Type_1099 is NULL.	SB-POIMP-20092
Perform_Other_Validations/ Check_Unit_Price	Checks if Not_To_Exceed_Price is greater than or equal to Unit_Price for Document Types STANDARD, BLANKET.	SB-POIMP-20067
Perform_Other_Validations/ Check_Unit_Price_For_Null	Checks if Unit_Price is NULL or lesser than Zero.	SB-POIMP-20087
Perform_Other_Validations/ Chk_Alw_Prc_Ovd_Flag_For_Quot	Checks if Allow_Price_Override_Flag is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20072
Perform_Other_Validations/ Chk_Freight_Carrier_For_Blkt	Checks if Freight_Carrier is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20082
Perform_Other_Validations/ Chk_Neg_By_Prep_Flag_For_Quot	Checks if Negotiated_By_Preparer_Flag is NULL if Header Document Type is 'QUOTATION'.	SB-POIMP-20074
Perform_Other_Validations/ Chk_Qty_Rcv_Exc_Code_For_Blkt	Checks if Qty_Rcv_Exception_Code is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20081
Perform_Other_Validations/ Chk_Qty_Rcv_Tol_For_Blkt	Checks if Qty_Rcv_Tolerance is NULL if Header Document Type is 'BLANKET'.	SB-POIMP-20080

Table 23 sb_validate_poi_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate	This is the main validation package for the Interface Lines Table (sb_po_lines_interface).	N/A
Validate_Foreign_Key_Checks/ Check_From_Header_ID	Checks if From_Header_Id is a valid value in the PO_HEADERS_ALL table.	SB-POIMP-20044
Validate_Foreign_Key_Checks/ Check_Category_ID	Checks if Category_Id is a valid value in the MTL_CATEGORIES_B table.	SB-POIMP-20038
Validate_Foreign_Key_Checks/ Check_Closed_By	Checks if Closed_By is a valid value in the HR_EMPLOYEES table.	SB-POIMP-20043
Validate_Foreign_Key_Checks/ Check_Fob_Active	Checks whether FOB is Active or Inactive.	SB-POIMP-25014
Validate_Foreign_Key_Checks/ Check_Freight_Terms_Active	Checks whether Freight_Terms is Active or Inactive.	SB-POIMP-25015
Validate_Foreign_Key_Checks/ Check_From_Line_ID	Checks if From_Line_Id is a valid value in the PO_LINES_ALL.	SB-POIMP-20045
Validate_Foreign_Key_Checks/ Check_Hazard_Class	Checks if Hazard_Class is a valid value.	SB-POIMP-20059
Validate_Foreign_Key_Checks/ Check_Hazard_Class_ID	Checks if Hazard_Class_Id is a valid value in the PO_HAZARD_CLASSES_B table.	SB-POIMP-20041
Validate_Foreign_Key_Checks/ Check_Item	Checks if Item is a valid value.	SB-POIMP-20052
Validate_Foreign_Key_Checks/ Check_Item_Id	Checks if Item_Id is a valid value in the MTL_SYSTEM_ITEMS_B table.	SB-POIMP-20050
Validate_Foreign_Key_Checks/ Check_Item_Revision	Checks if Item_Revision is a valid value in the MTL_ITEM_REVISIONS table.	SB-POIMP-20047
Validate_Foreign_Key_Checks/ Check_Line_Num	Checks if Check_Line_Num is a valid value.	SB-POIMP-20056
Validate_Foreign_Key_Checks/ Check_Line_Type	Checks if Check_Line_Type is a valid value.	SB-POIMP-20055
Validate_Foreign_Key_Checks/ Check_Line_Type_ID	Checks if Line_Type_Id is a valid value in the PO_LINE_TYPES_B table.	SB-POIMP-20039
Validate_Foreign_Key_Checks/ Check_Receiving_Routing	Checks if Receiving_Routing is valid.	SB-POIMP-20064
Validate_Foreign_Key_Checks/ Check_Ship_To_Location_Code	Checks if Ship_To_Location is a valid value.	SB-POIMP-20061
Validate_Foreign_Key_Checks/ Check_Ship_To_Org_Code	Checks if Ship_To_Organization_Code is a valid value.	SB-POIMP-20060
Validate_Foreign_Key_Checks/ Check_Shipment_Num	Checks if Shipment_Num is a valid value.	SB-POIMP-20057

Table 23 sb_validate_poi_lines_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Tax_Code_Id	Check if Tax_Code_Id is valid.	SB-POIMP-20065
Validate_Foreign_Key_Checks/ Check_Tax_Name	Checks if Tax_Name is a valid value in the AP_TAX_CODES_ALL table.	SB-POIMP-20046
Validate_Foreign_Key_Checks/ Check_Template_Name	Checks if Template_Name is a valid value.	SB-POIMP-20062
Validate_Foreign_Key_Checks/ Check_Un_Number	Checks if Un_Number is a valid value.	SB-POIMP-20058
Validate_Foreign_Key_Checks/ Check_Un_Number_ID	Checks if Un_Number_Id is a valid value in the PO_UN_NUMBERS_B table.	SB-POIMP-20040
Validate_Foreign_Key_Checks/ Check_Unit_Of_Measure	Checks if Unit_Of_Measure is a valid value in the MTL_UNITS_OF_MEASURE_TL table.	SB-POIMP-20048
Validate_Foreign_Key_Checks/ Check_Unit_Of_Measure_Org	Checks if the Unit_Of_Measure is a valid value for a particular Item and Organization in the MTL_ITEM_UOMS_VIEW view.	SB-POIMP-20049
Validate_Foreign_Key_Checks/ Check_Uom_Code	Checks if Uom Code is a valid value.	SB-POIMP-20054
Validate_Foreign_Key_Checks/ Check_Ussgl_Transaction_Code	Checks if Ussgl_Transaction_Code is a valid value in the GL_USSGL_TRANSACTION_CODES table.	SB-POIMP-20042
Validate_Required_Columns/ Check_Interface_Header_ID	Checks the Interface Header ID column. It should be entered since it is a mandatory Column.	SB-POIMP-20034
Validate_Required_Columns/ Check_Interface_Line_ID	Checks the Interface Line ID column. It should be entered since it is a mandatory Column.	SB-POIMP-20035
Validate_Required_Columns/ Check_Organization_ID	Checks the Organization ID column. It should be entered since it is a mandatory Column.	SB-POIMP-20036

A.2.2 Receiving

Master

Table 24 sb_validate_rcv_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate	This is the main validation procedure of the Receiving Open Interface.	SB-POREC-25006

Header

Table 25 sb_validate_rcv_hdr_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Check_For_Errors	Checks if an error has occurred by checking the count of updated records for errors.	None
Derive_Required_Columns/ Check_ASN_TYPE	Validates for 'ASN' or 'ASBN' values for ASN_TYPE in rcv_headers_interface.	SB-POREC20107
Derive_Required_Columns/ Check_Auto_Transact_Code	Checks for valid values in Auto_Transact_Code field of rcv_headers_interface table. The values allowed are 'SHIP', 'RECEIVE', 'DELIVER' when ASN_TYPE is 'ASN' for a given Employee_Id or Employee_Name.	SB-POREC20112
Derive_Required_Columns/ Check_Employee_Name_Id	Checks for a valid value in Employee_Id or Employee_Name field of rcv_header_interface table. Any one of the fields should not be null when Auto_Transact_Code is 'RECEIVE'.	SB-POREC20118
Derive_Required_Columns/ Check_Invoice_Date	Checks for a valid value in Invoice_Date field of rcv_header_interface table. The field should not be null when ASN_TYPE is 'ASBN'.	SB-POREC20116
Derive_Required_Columns/ Check_Invoice_Num	Checks for a valid value in Invoice_Num field of rcv_header_interface table. The field should not be null when ASN_TYPE is 'ASBN'.	SB-POREC20115
Derive_Required_Columns/ Check_Processing_Status_Code	Validates for 'PENDING' value for Processing_Status_Code in rcv_headers_interface.	SB-POREC20108
Derive_Required_Columns/ Check_Receipt_Num	Checks for a valid value Receipt_Num field of rcv_headers_interface table. There should be a value when Auto_Transact_Code in 'SHIP'.	SB-POREC20113

Table 25 sb_validate_rcv_hdr_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Check_Receipt_Source_Code	Validates for 'VENDOR' value for Receipt_Source_Code in rcv_headers_interface.	SB-POREC20109
Derive_Required_Columns/ Check_Shipped_Date	Checks for a valid value in Shipped_Date field of rcv_header_interface table. The field should not be null when ASN_TYPE is either 'ASN' or 'ASBN' and it should be earlier than current date and Expected_Receipt_Date.	SB-POREC20114
Derive_Required_Columns/ Check_Shipped_Date_Range	Checks the SHIPPED_DATE value with the current date and EXPECTED_RECEIPT_DATE. It should be earlier than or equal to current date and EXPECTED_RECEIPT_DATE.	SB-POREC20189
Derive_Required_Columns/ Check_Total_Invoice_Amount	Checks for a valid value in Total_Invoice_Amount field of rcv_header_interface table. The field should not be null when ASN_TYPE is 'ASBN' and it should not be a negative value.	SB-POREC20117
Derive_Required_Columns/ Check_Transaction_Type	Checks for valid values in Transaction_Type field of rcv_headers_interface table. The values allowed are 'NEW' or 'CANCEL'.	SB-POREC20110
Derive_Required_Columns/ Check_Validation_Flag	Checks for valid values in Validation_Flag field of rcv_headers_interface table. The values allowed are 'Y' or 'N'.	SB-POREC20111
Derive_Required_Columns/ Check_Validity_Employee_Name	Checks the validity of employee_name with the corresponding column in hr_employees.	SB-POREC20190
Derive_Required_Columns/ Check_Vendor_Name	Checks for the validity of Vendor_Name in po_suppliers_val_v.	SB-POREC20119
Derive_Required_Columns/ Derive_Employee_Id	Derives Employee_Id for a given Employee_Name.	N/A
Derive_Required_Columns/ Derive_Location_Id	Derives Location_Id for a given Location_Code.	N/A
Derive_Required_Columns/ Derive_Org_ID_From_Code	Derives Ship_To_Organization_ID for a given Ship_To_Organization_Code.	N/A
Derive_Required_Columns/ Derive_Payment_Terms_ID	Derives Payment_Terms_Id for a given Payment_Terms_Name.	N/A
Derive_Required_Columns/ Derive_Vendor_ID	Derives Vendor_Id for a given Vendor_Name and Vendor_Num.	N/A
Derive_Required_Columns/ Derive_Vendor_Site_ID	Derives Vendor_Site_Id for a given Vendor_Id and Vendor_Site_Code.	N/A

Table 25 sb_validate_rcv_hdr_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate	Performs all the pre-validations pertaining to Purchase Order Headers Interface table.	N/A
Validate_Required_Columns/ Check_Group_ID	Validates for the Not null value of Group_ID in rcv_headers_interface table.	SB-POREC20101
Validate_Required_Columns/ Check_Header_Interface_ID	Validates for the Not null value of Header_Interface_ID in rcv_headers_interface table.	SB-POREC20100
Validate_Required_Columns/ Check_Processing_Status_Code	Validates for the Not Null value of Processing_Status_Code in rcv_headers_interface table.	SB-POREC20102
Validate_Required_Columns/ Check_Receipt_Source_Code	Validates for the Not Null value of Receipt_Source_Code in rcv_headers_interface table.	SB-POREC20103
Validate_Required_Columns/ Check_Transaction_Type	Validates for the Not Null value of transaction_type in rcv_headers_interface table.	SB-POREC20104
Validate_Required_Columns/ Check_Validation_Flag	Validates for the Not Null value of validation_flag.	SB-POREC20106
Validate_Required_Columns/ Check_Vendor_Name_Num	Validates for the Not Null value of vendor_name or vendor_num in rcv_headers_interface table. Either one value should be entered.	SB-POREC20105

Transactions

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Check_For_Errors	Checks if an error has occurred by checking the count of updated records for errors.	None
Validate_Required_Columns/ Check_Auto_Transact_Code	Checks for a valid value in Auto_Transact_Code field of rcv_transactions_header table.	SB-POREC20131
Derive_Required_Columns/ Check_Auto_Transact_Code	Validates for a valid value in Auto_Transact_Code field in rcv_transactions_interface table. The values allowed are 'RECEIVE', 'DELIVER', and 'SHIP'.	SB-POREC20177
Validate_Foreign_Key_Checks/ Check_Bom_Resource_Id	Does the foreign key validation for bom_resource_id field in rcv_transactions_interface table.	SB-POREC20160
Validate_Foreign_Key_Checks/ Check_Charge_Account_Id	Does the foreign key validation for charge_account_id field in rcv_transaction_interface table.	SB-POREC20151
Validate_Foreign_Key_Checks/ Check_Currency_Conversion_Type	Does the foreign key validation for currency_conversion_type field in rcv_transactions_interface table.	SB-POREC20166
Validate_Foreign_Key_Checks/ Check_Department_Code	Does the foreign key validation for department_code field in rcv_transactions_interface table.	SB-POREC20161
Derive_Required_Columns/ Check_Destination_Type_Code	Checks for valid destination_type_code when auto_transaction_code is 'DELIVER'.	SB-POREC20186
Validate_Required_Columns/ Check_Document_Line_Num	Checks for a valid value in Document_Line_Num, Item_Num, Vendor_Item_Num, Item_Id or Po_Line_Id field of rcv_transactions_header table.	SB-POREC20130
Validate_Required_Columns/ Check_Document_Num	Checks for a valid value in Document_Num or Po_Header_Id field of rcv_transactions_header table. Either one should have a valid value.	SB-POREC20136
Validate_Foreign_Key_Checks/ Check_Employee_Id	Does the foreign key validation for employee_id field in rcv_transaction_interface table.	SB-POREC20152
Derive_Required_Columns/ Check_Employee_Id	Checks for valid values in employee_id when the transaction_type is 'DELIVER'.	SB-POREC20182

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Employee_Id1	Does the foreign key validation for employee_id field in rcv_transactions_interface table.	SB-POREC20152
Derive_Required_Columns/ Check_Expected_Receipt_Date	Checks for availability of expected_receipt_date in rcv_headers_interface table.	SB-POREC20187
Validate_Foreign_Key_Checks/ Check_Freight_Carrier_Code	Does the foreign key validation for freight_carrier_code field in rcv_transactions_interface table.	SB-POREC20162
Validate_Required_Columns/ Check_Group_ID	Checks for a valid value in Group_ID field of rcv_transactions_header table.	SB-POREC20121
Derive_Required_Columns/ Check_Group_Id	Checks for valid group_id in rcv_headers_interface table.	SB-POREC20183
Derive_Required_Columns/ Check_Header_Interface_Id	Checks and validates for the presence of valid Header_Interface_Id in rcv_headers_interface table.	SB-POREC20180
Validate_Required_Columns/ Check_Header_Interface_Id	Checks for a valid value in Header_Interface_Id field of rcv_transactions_header table.	SB-POREC20137
Validate_Foreign_Key_Checks/ Check_Inspection_Quality_Code	Does the foreign key validation for inspection_quality_code field in rcv_transaction_interface table.	SB-POREC20150
Validate_Required_Columns/ Check_Interface_Transaction_ID	Checks for a valid value in Interface_Transaction_ID field of rcv_transactions_header table.	SB-POREC20120
Validate_Foreign_Key_Checks/ Check_Inv_Transaction_Id	Does the foreign key validation for Inv_Transaction_Id field in rcv_transaction_interface table.	SB-POREC20145
Derive_Required_Columns/ Check_Item_Catgry	Validates for a valid value in ITEM_CATEGORY or CATEGORY_ID or DOCUMENT_LINE_NUM or PO_LINE_ID fields in rcv_transactions_interface table. At least one value should be provided.	SB-POREC20173
Validate_Required_Columns/ Check_Item_Description	Checks for a valid value in Item_Description field of rcv_transactions_header table.	SB-POREC20129
Validate_Foreign_Key_Checks/ Check_Item_Id	Does the foreign key validation for item_id field in rcv_transaction_interface table.	SB-POREC20154
Validate_Foreign_Key_Checks/ Check_Locator_Id	Does the foreign key validation for locator_id field in rcv_transactions_interface table.	SB-POREC20164

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Check_Org_Code_Id	Checks for TO_ORGANIZATION_CODE or TO_ORGANIZATION_ID (at least one value should be specified).	SB-POREC20185
Validate_Foreign_Key_Checks/ Check_Parent_Transaction_ID	Does the foreign key validation for Parent_Transaction_Id in rcv_transaction_interface table.	SB-POREC20141
Validate_Foreign_Key_Checks/ Check_Po_Distribution_Id	Does the foreign key validation for Po_Distribution_Id field in rcv_transaction_interface table.	SB-POREC20146
Validate_Foreign_Key_Checks/ Check_Po_Header_Id	Does the foreign key validation for Po_Header_Id field in rcv_transaction_interface table.	SB-POREC20142
Validate_Foreign_Key_Checks/ Check_Po_Line_Id	Does the foreign key validation for Po_Line_Id field in rcv_transaction_interface table.	SB-POREC20143
Validate_Foreign_Key_Checks/ Check_Po_Line_Location_Id	Does the foreign key validation for Po_Line_Location_Id field in rcv_transaction_interface table.	SB-POREC20144
Validate_Foreign_Key_Checks/ Check_Po_Release_Id	Does the foreign key validation for Po_Release_Id field in rcv_transaction_interface table.	SB-POREC20147
Validate_Foreign_Key_Checks/ Check_Primary_Unit_Of_Measure	Does the foreign key validation primary_unit_of_measure field in rcv_transactions_interface table.	SB-POREC20172
Validate_Required_Columns/ Check_Processing_Mode_Code	Checks for a valid value in Processing_Mode_Code field of rcv_transactions_header table.	SB-POREC20125
Derive_Required_Columns/ Check_Processing_Mode_Code	Validates for a valid value in Processing_Mode_Code field in rcv_transactions_interface table. The value allowed is 'BATCH'.	SB-POREC20176
Derive_Required_Columns/ Check_Processing_Status_Code	Validates for a valid value in Processing_Status_Code field in rcv_transactions_interface table. The value allowed is 'PENDING'.	SB-POREC20175
Validate_Required_Columns/ Check_Processing_Status_Code	Checks for a valid value in Processing_Status_Code field of rcv_transactions_header table.	SB-POREC20124
Validate_Required_Columns/ Check_Quantity	Checks for a valid value in Quantity field of rcv_transactions_header table. The value should not be negative.	SB-POREC20127

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Reason_Id	Does the foreign key validation for reason_id field in rcv_transactions_interface table.	SB-POREC20167
Validate_Required_Columns/ Check_Receipt_Source_Code	Checks for a valid value in Receipt_Source_Code field of rcv_transactions_header table.	SB-POREC20132
Derive_Required_Columns/ Check_Receipt_Source_Code	Validates for a valid value in Receipt_Source_Code field in rcv_transactions_interface table. The value allowed is 'VENDOR'.	SB-POREC20178
Validate_Foreign_Key_Checks/ Check_Req_Distribution_Id	Does the foreign key validation for Requisition_Distribution_Id field in rcv_transaction_interface table.	SB-POREC20149
Validate_Foreign_Key_Checks/ Check_Requisition_Line_Id	Does the foreign key validation for Requisition_Line_Id field in rcv_transaction_interface table.	SB-POREC20148
Validate_Foreign_Key_Checks/ Check_Routing_Header_ID	Does the foreign key validation for Routing_Header_Id in rcv_transactions_interface table.	SB-POREC20139
Validate_Foreign_Key_Checks/ Check_Routing_Step_ID	Does the foreign key validation for Routing_Step_Id in rcv_transactions_interface table.	SB-POREC20140
Derive_Required_Columns/ Check_Ship_Code_Id	Checks for availability of valid values for SHIP_TO_LOCATION_CODE or SHIP_TO_LOCATION_ID when the ASN_TYPE is 'ASN'.	SB-POREC20184
Validate_Foreign_Key_Checks/ Check_Shipment_Header_Id	Does the foreign key validation shipment_header_id field in rcv_transactions_interface table.	SB-POREC20170
Validate_Foreign_Key_Checks/ Check_Shipment_Line_Id	Does the foreign key validation shipment_line_id field in rcv_transactions_interface table.	SB-POREC20171
Validate_Foreign_Key_Checks/ Check_Source_Doc_Unit_Of_Measr	Does the foreign key validation for source_doc_unit_of_measure field in rcv_transactions_interface table.	SB-POREC20168
Validate_Required_Columns/ Check_Source_Document_Code	Checks for a valid value in Source_Document_Code field of rcv_transactions_header table.	SB-POREC20135
Derive_Required_Columns/ Check_Source_Document_Code	Validates for a valid value in Source_Document_Code field in rcv_transactions_interface table. The value allowed is 'PO'.	SB-POREC20179

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Subinventory	Does the foreign key validation for subinventory field in rcv_transactions_interface table.	SB-POREC20163
Validate_Required_Columns/ Check_Transaction_Date	Checks for a valid value in Transaction_Date field of rcv_transactions_header table.	SB-POREC20123
Validate_Required_Columns/ Check_Transaction_Status_Code	Checks for a valid value in Transaction_Status_Code field of rcv_transactions_header table.	SB-POREC20126
Derive_Required_Columns/ Check_Transaction_Status_Code	Checks for value in Transaction_Status_Code. Values allowed are 'ERROR' and 'COMPLETED'.	SB-POREC-20188
Validate_Required_Columns/ Check_Transaction_Type	Checks for a valid value in Transaction_Type field of rcv_transactions_header table.	SB-POREC20122
Derive_Required_Columns/ Check_Transaction_Type	Validates for a valid value in Transaction_Type field in rcv_transactions_interface table. The values allowed are 'SHIP' or 'RECEIVE'.	SB-POREC20174
Validate_Foreign_Key_Checks/ Check_Tsrptn_Account_Id	Does the foreign key validation for Transportation_Account_Id field in rcv_transactions_interface table.	SB-POREC20155
Validate_Required_Columns/ Check_Unit_Of_Measure	Checks for a valid value in Unit_Of_Measure field of rcv_transactions_header table.	SB-POREC20128
Validate_Foreign_Key_Checks/ Check_UOM_Code	Does the foreign key validation for uom_code field in rcv_transactions_interface table.	SB-POREC20165
Derive_Required_Columns/ Check_Valid_Item_Category	Checks the validity of Item_category in mtl_categories table.	SB-POREC-20191
Derive_Required_Columns/ Check_Valid_Loc_Code	Checks validity of ship_to_location_code in hr_locations table.	SB-POREC-20192
Derive_Required_Columns/ Check_Valid_Vendor_Name	Checks validity of Vendor_Name in po_suppliers_val_v table.	SB-POREC-20193
Derive_Required_Columns/ Check_Valid_Vendor_Num	Checks validity of Vendor_Num in po_suppliers_val_v table.	SB-POREC-20194
Validate_Required_Columns/ Check_Validation_Flag	Checks for a value in Header_Interface_Id field of rcv_transactions_header table. Should not be null.	SB-POREC20138

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Check_Validation_Flag	Checks for valid values in validation_flag in rcv_transactions_interface table; accepts 'Y' or 'N'.	SB-POREC20181
Validate_Required_Columns/ Check_Vendor_Name_Num	Checks for a valid value in Vendor_Name or Vendor_Num field of rcv_transactions_header table. Either one should have a valid value.	SB-POREC20134
Validate_Foreign_Key_Checks/ Check_Wip_Entity_Id	Does the foreign key validation for wip_entity_id field in rcv_transaction_interface table.	SB-POREC20153
Validate_Foreign_Key_Checks/ Check_Wip_Line_Id	Does the foreign key validation for wip_line_id field in rcv_transactions_interface table.	SB-POREC20156
Validate_Foreign_Key_Checks/ Check_Wip_Rptv_Schedule_Id	Does the foreign key validation for repetitive_schedule_id field in rcv_transactions_interface table.	SB-POREC20157
Validate_Foreign_Key_Checks/ Check_Worwip_Entity_Id	Does the foreign key validation for wip_entity_id field in rcv_transactions_interface with wip_operations_resources table.	SB-POREC20159
Validate_Foreign_Key_Checks/ Check_Wowip_Entity_Id	Does the foreign key validation for wip_entity_id field in rcv_transactions_interface with wip_operations table.	SB-POREC20158
Derive_Required_Columns/ Derive_Category_Id	Derives Category_Id for a given Item_Category.	N/A
Derive_Required_Columns/ Derive_Deliver_to_Location_ID	Derives Deliver_to_Location_ID for a given Deliver_to_Location_Code.	N/A
Derive_Required_Columns/ Derive_Distribution_Num	Derives Distribution_Num for a given po_distribution_id.	N/A
Derive_Required_Columns/ Derive_Document_Line_Num	Derives Document_Line_Num for a given po_line_id.	N/A
Derive_Required_Columns/ Derive_Document_Num	Derives Document_Num for a given Po_Header_Id.	N/A
Derive_Required_Columns/ Derive_Employee_ID	Derives Employee_ID for a given Employee_Name when transaction_Type is 'DELIVER'.	N/A
Derive_Required_Columns/ Derive_Frieght_Carrier_Code	Derives Frieght_Carrier_Code for a given organization_id.	N/A
Derive_Required_Columns/ Derive_Location_ID	Derives a Location_ID for a given Location_Code.	N/A

Table 26 sb_validate_rcv_trans_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ Derive_Release_num	Derives Release_Num for a given po_release_id.	N/A
Derive_Required_Columns/ Derive_Ship_Loc_Code_ID	Derives a Ship_To_Location_Id for a given Ship_To_Location_Code.	N/A
Derive_Required_Columns/ Derive_To_Org_id	Derives To_Organization_ID for a given to_organization_Code.	N/A
Derive_Required_Columns/ Derive_Vendor_Site_ID	Derives Vendor_Site_ID for a given Vendor_Site_Code.	N/A
Derive_Required_Columns/ Derive_Vendor_ID	Derives Vendor_Id for a given Vendor_Name and Vendor_Num.	N/A

A.2.3 Requisitions

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Derive_Required_Columns/ accrual_acc_ID_Exp	Derives Accrual_Account_ID from po_system_parameters table if Destination_Type_Code EXPENSE.	N/A
Derive_Required_Columns/ accrual_acc_ID_Inv	Derives Accrual_Account_ID from mtl_parameters table if Destination_Type_Code is INVENTORY.	N/A
Derive_Required_Columns/ Budget_Acc_ID_Exp	Derives Budget_Account_ID from Charge_Account_ID field if Destination_Type_Code is EXPENSE or SHOP FLOOR.	N/A
Derive_Required_Columns/ Budget_Acc_ID_Inv	Derives Budget_Account_ID from mtl_item_sub_inventories, mtl_secondary_inventories tables if Destination_Type_Code is INVENTORY.	N/A
Derive_Required_Columns/ Item_Description	Derives Item_Description for a given Item_ID from Mtl_System_Items_B table.	N/A
Derive_Required_Columns/ Unit_Price	Derives Unit_Price for a given Item_ID from Mtl_System_Items table.	N/A
Derive_Required_Columns/ Variance_Acc_ID_Exp	Derives Variance_Account_ID from Charge_Account_ID field if Destination_Type_Code is EXPENSE.	N/A
Derive_Required_Columns/ Variance_Acc_ID_Inv	Derives Variance_Account_ID from mtl_parameters table if Destination_Type_Code is INVENTORY or SHOP FLOOR.	N/A
Perform_Other_Validations/ Check_Category_ID_Val	Checks the validity of Category_ID for the given Item_ID and Organization_ID.	SB-POREQ-20171
Perform_Other_Validations/ Check_Deli_To_Loc_ID_FK	Checks validity of Deliver_To_Location_ID in Hr_Locations table.	SB-POREQ-20170
Perform_Other_Validations/ Check_Dest_Type_Code_Exp	Checks whether the item is Purchasing enabled for a given Destination_Organization_ID if Destination_Type_Code is EXPENSE.	SB-POREQ-20166
Perform_Other_Validations/ Check_Dest_Type_Code_Inv	Checks whether the item is stock enabled for a given Destination_Organization_ID if Destination_Type_Code is INVENTORY.	SB-POREQ-20165

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Perform_Other_Validations/ Check_Dest_Type_Code_Shop	Checks whether the item is Purchasing enabled and Stock enabled for a given Destination_Organization_ID if Destination_Type_Code is SHOP FLOOR.	SB-POREQ-20167
Perform_Other_Validations/ Check_Src_Type_Code_Inv	Checks whether the item is Stock enabled and Internal Order enabled for a given Source_Organization_ID if Source_Type_Code is INVENTORY.	SB-POREQ-20168
Perform_Other_Validations/ Check_Src_Type_Code_Ven	Checks whether the item is purchasing enabled given Destination_Organization_ID if Source_Type_Code is VENDOR.	SB-POREQ-20169
Validate_Foreign_Key_Checks/ Check_Accrual_Account_ID	Checks validity of Accrual_Account_ID in Gl_Code_Combinations table.	SB-POREQ-20138
Validate_Foreign_Key_Checks/ Check_Approver_ID	Checks validity of Approver_ID in hr_employees table.	SB-POREQ-20145
Validate_Foreign_Key_Checks/ Check_Autosource_Doc_Header_ID	Checks validity of Autosource_Doc_Header_ID in po_autosource_documents_all table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20146
Validate_Foreign_Key_Checks/ Check_Budget_Account_ID	Checks validity of Budget_Account_ID in Gl_Code_Combinations table.	SB-POREQ-20140
Validate_Foreign_Key_Checks/ Check_Category_ID	Checks validity of Category_ID in Mtl_Categories_B table.	SB-POREQ-20116
Validate_Foreign_Key_Checks/ Check_Charge_Account_ID_FK	Checks validity of Charge_Account_ID in gl_code_combinations table.	SB-POREQ-20164
Validate_Foreign_Key_Checks/ Check_Currency_Code	Checks validity of Currency_Code in fnd_currencies table.	SB-POREQ-20142
Validate_Foreign_Key_Checks/ Check_Del_To_Req_ID_FK	Checks validity of Deliver_To_Requestor ID in hr_employees table.	SB-POREQ-20131
Validate_Foreign_Key_Checks/ Check_Dest_Org_ID_FK	Checks validity of Destination_Organization_ID in org_organization_definitions table.	SB-POREQ-20128
Validate_Foreign_Key_Checks/ Check_Dest_Subin_FK	Checks validity of Destination_Subinventory for a given Destination_Organization_ID in MTL_Secondary_Inventories table.	SB-POREQ-20130
Validate_Foreign_Key_Checks/ Check_Document_Type_Code	Checks validity of Document_Type_Code in po_document_types_all_b table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20159

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Expenditure_Type	Checks validity of Expenditure_Type in pa_expenditure_types table.	SB-POREQ-20158
Validate_Foreign_Key_Checks/ Check_HAZARD_CLASS_ID	Checks validity of Hazard_Class_ID in po_hazard_classes_b table.	SB-POREQ-20120
Validate_Foreign_Key_Checks/ Check_Item_ID	Checks validity of Item_ID in mtl_system_items_b table.	SB-POREQ-20162
Validate_Foreign_Key_Checks/ Check_Item_Revision	Checks validity of Item_Revision in mtl_item_revisions table.	SB-POREQ-20163
Validate_Foreign_Key_Checks/ Check_Kanban_Card_ID	Checks validity of Kanban_Card_ID in mtl_kanban_cards table.	SB-POREQ-20160
Validate_Foreign_Key_Checks/ Check_Line_Type_ID	Checks validity of Line_Type in po_line_types_b table.	SB-POREQ-20118
Validate_Foreign_Key_Checks/ Check_Note1_ID	Checks validity of Note1_ID in Po_Notes table.	SB-POREQ-20137
Validate_Foreign_Key_Checks/ Check_Note2_ID	Checks validity of Note2_ID in Po_Notes table.	SB-POREQ-20149
Validate_Foreign_Key_Checks/ Check_Note3_ID	Checks validity of Note3_ID in Po_Notes table.	SB-POREQ-20150
Validate_Foreign_Key_Checks/ Check_Note4_ID	Checks validity of Note4_ID in Po_Notes table.	SB-POREQ-20151
Validate_Foreign_Key_Checks/ Check_Note5_ID	Checks validity of Note5_ID in Po_Notes table.	SB-POREQ-20152
Validate_Foreign_Key_Checks/ Check_Note6_ID	Checks validity of Note6_ID in Po_Notes table.	SB-POREQ-20153
Validate_Foreign_Key_Checks/ Check_Note7_ID	Checks validity of Note7_ID in Po_Notes table.	SB-POREQ-20154
Validate_Foreign_Key_Checks/ Check_Note8_ID	Checks validity of Note8_ID in Po_Notes table.	SB-POREQ-20155
Validate_Foreign_Key_Checks/ Check_Note9_ID	Checks validity of Note9_ID in Po_Notes table.	SB-POREQ-20156
Validate_Foreign_Key_Checks/ Check_Note10_ID	Checks validity of Note10_ID in Po_Notes table.	SB-POREQ-20157
Validate_Foreign_Key_Checks/ Check_Preparer_ID_FK	Checks validity of Preparer_ID in hr_employees table.	SB-POREQ-20161
Validate_Foreign_Key_Checks/ Check_Project_ID	Checks validity of Project_ID in pa_projects_all table.	SB-POREQ-20147
Validate_Foreign_Key_Checks/ Check_Rate	Checks validity of Rate in gl_daily_conversion_rates_r10 table.	SB-POREQ-20143
Validate_Foreign_Key_Checks/ Check_Rate_Type	Checks validity of Rate_Type in gl_daily_conversion_types table.	SB-POREQ-20144

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_Req_Distribution_ID	Checks validity of Req_Distribution_ID in Po_Req_Distributions_All table	SB-POREQ-20135
Validate_Foreign_Key_Checks/ Check_Requisition_Header_ID	Checks validity of Requisition_Header_ID in PO_Requisition_Headers_All table.	SB-POREQ-20115
Validate_Foreign_Key_Checks/ Check_Requisition_Line_ID	Checks validity of Requisition_Line_ID in po_requisition_lines_all table.	SB-POREQ-20126
Validate_Foreign_Key_Checks/ Check_Source_Organization_ID	Checks validity of Source_Organization_ID in org_organization_definitions table, if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20127
Validate_Foreign_Key_Checks/ Check_Source_Subinventory	Checks validity of Source_Subinventory in MTL_Secondary_Inventories table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20129
Validate_Foreign_Key_Checks/ Check_Sug_Vendor_Con_ID	Checks validity of Suggested_Vendor_Contact_ID in Po_Vendor_Contacts table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20136
Validate_Foreign_Key_Checks/ Check_Suggested_Buyer_ID	Checks validity of Suggested_Buyer_ID in po_agents table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20132
Validate_Foreign_Key_Checks/ Check_Suggested_Vendor_ID	Checks validity of Suggested_Vendor_ID in po_vendors table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20133
Validate_Foreign_Key_Checks/ Check_Suggested_Vendor_Site_ID	Checks validity of Suggested_Vendor_Site_ID in po_vendor_sites_all table if Auto source Flag is not equal to 'Y' or 'P'.	SB-POREQ-20134
Validate_Foreign_Key_Checks/ Check_Task_ID	Checks validity of Task_ID in pa_tasks table.	SB-POREQ-20148
Validate_Foreign_Key_Checks/ Check_UN_Number_ID	Checks validity of UN_Number_ID in po_un_numbers_b table.	SB-POREQ-20119
Validate_Foreign_Key_Checks/ Check_Unit_Of_Measure	Checks validity of Unit_Of_Measure in mtl_units_of_measure_tl table.	SB-POREQ-20117
Validate_Foreign_Key_Checks/ Check_Ussgl_Transaction_Code	Checks validity of Ussgl_Transaction_Code in gl_ussgl_transaction_codes table.	SB-POREQ-20141
Validate_Foreign_Key_Checks/ Check_Variance_Account_ID	Checks validity of Variance_Account_ID in Gl_Code_Combinations table.	SB-POREQ-20139

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Foreign_Key_Checks/ Check_WIP_Entity_ID	Checks validity of WIP_Entity_ID for a given Destination_Organization_D in wip_entities table.	SB-POREQ-20121
Validate_Foreign_Key_Checks/ Check_WIP_Line_ID	Checks validity of WIP_Line_ID for a given Destination_Organization_ID in wip_lines table.	SB-POREQ-20122
Validate_Foreign_Key_Checks/ Check_WIP_Entity_ID_O	Checks validity of WIP_Entity_ID for a given Destination_Organization_ID, Operation_Sequence_Num, and Repetitive_Schedule_ID in wip_operations table.	SB-POREQ-20123
Validate_Foreign_Key_Checks/ Check_WIP_Entity_ID_OR	Checks validity of WIP_Entity_ID for a given Destination_Organization_ID, Operation_Sequence_Num, and Repetitive_Schedule_ID in wip_operation_resources table.	SB-POREQ-20124
Validate_Foreign_Key_Checks/ Check_WIP_Rept_Sch_ID	Checks validity of WIP_Repetitive_Schedule_ID for a given Destination_Organization_ID in wip_repetitive_schedules table.	SB-POREQ-20125
Validate_Required_Columns/ Check_Authorization_Status	Checks for the required field Authorization_Status in sb_po_req_interface_all table.	SB-POREQ-20102
Validate_Required_Columns/ Check_Charge_Account_ID	Checks for the required fields Charge_Account_ID and Charge_Account_Segment in sb_po_req_interface_all table.	SB-POREQ-20105
Validate_Required_Columns/ Check_Currency_Code	Checks for the Rate_Date and Rate_Type if Currency_Code exists in sb_po_req_interface_all table.	SB-POREQ-20109
Validate_Required_Columns/ Check_Deli_To_Loc_ID	Checks for the required Deliver_To_Location_ID and Deliver_To_Location_Code in sb_po_req_interface_all table.	SB-POREQ-20107
Validate_Required_Columns/ Check_Deli_To_Reqtor_ID	Checks for the required Deliver_To_Reqstor_ID and Deliver_To_Reqstor_Name in sb_po_req_interface_all table.	SB-POREQ-20108
Validate_Required_Columns/ Check_Dest_Org_ID	Checks for the required fields Destination_Organization_ID and Destination_Organization_Code in sb_po_req_interface_all table.	SB-POREQ-20106

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Required_Columns/ Check_Dest_Subin	Checks for the Item_ID and Item_Segment if Destination_Type_Code is INVENTORY in sb_po_req_interface_all table.	SB-POREQ-20111
Validate_Required_Columns/ Check_Dest_Subinv	Checks for Destination_Subinventory if Destination_Type_Code is INVENTORY in sb_po_req_interface_all table.	SB-POREQ-20111
Validate_Required_Columns/ Check_Destination_Type_Code	Checks for the required field Destination_Type_Code in sb_po_req_interface_all table.	SB-POREQ-20101
Validate_Required_Columns/ Check_Interface_Source_Code	Checks for the required field Interface_Source_Code in sb_po_req_interface_all table.	SB-POREQ-20100
Validate_Required_Columns/ Check_Preparer_ID	Checks for the required fields Preparer_ID and Preparer_Name in sb_po_req_interface_all table.	SB-POREQ-20104
Validate_Required_Columns/ Check_Proj_Acct_Cont	Checks for the Project_Num, Project_ID, Task_ID, Task_Num, Expenditure_Type and Expenditure_Organization_ID if Project_Accounting_Context is 'Y' in sb_po_req_interface_all table.	SB-POREQ-20110
Validate_Required_Columns/ Check_Quantity	Checks for the required field Quantity in sb_po_req_interface_all table.	SB-POREQ-20103
Validate_Required_Columns/ Check_Req_Type	Checks for Requisition_Type is INTERNAL or PURCHASE in sb_po_req_interface_all table.	SB-POREQ-20114
Validate_Required_Columns/ Check_Src_Type_Code	Checks for the Item_ID and Item_Segment if Source_Type_Code is INVENTORY in sb_po_req_interface_all table.	SB-POREQ-20112
Validate_Required_Columns/ Check_Src_Type_Code1	Checks for Source Type Code if Destination Type Code is EXPENSE or SHOP FLOOR in sb_po_req_interface_all table.	SB-POREQ-20113
Validate_Required_Columns/ Deliver_To_Location_Id	Derives the Deliver_To_Location ID for a given Deliver_To_Location_Code in sb_po_req_interface_all table.	N/A
Validate_Required_Columns/ Deliver_To_Requestor_Id	Derives Deliver_To_Requestor_ID for a given Deliver_To_Requestor_Name in sb_po_req_interface_all table.	N/A

Table 27 sb_validate_por_req_pkg

Validation Procedure/ Sub-Routine	Purpose	Error Code
Validate_Required_Columns/ Derive_Desti_Org_ID	Derives Destination_Organization_ID for a given Destination_Organization_Code in sb_po_req_interface_all table.	N/A

Error Codes

B.1 Inventory

B.1.1 Customer Items

Table 28 sb_validate_cust_items_pkg

Error Code	Error Message	Generated from
SB-INVCIT-25301	Customer_Item_Number is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Cust_Item_Number
SB-INVCIT-25302	Transaction_Type is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Type
SB-INVCIT-25303	Item_Definition_Level is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Item_Def_Level
SB-INVCIT-25304	Customer Id, Number and Name are mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Customer
SB-INVCIT-25305	Address columns are mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Address
SB-INVCIT-25306	Customer category columns are mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Cust_Category
SB-INVCIT-25307	Commodity code columns are mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Commodity
SB-INVCIT-25308	Container item org columns are mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Container_Item
SB-INVCIT-25309	Model customer item columns are mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Model_Cust_Item
SB-INVCIT-25310	Customer_Id is invalid.	Validate_Foreign_Key_Checks/ Check_Customer_Id
SB-INVCIT-25311	Customer_Number is invalid.	Validate_Foreign_Key_Checks/ Check_Customer_No

Table 28 sb_validate_cust_items_pkg

Error Code	Error Message	Generated from
SB-INVCIT-25312	Customer_Name is invalid.	Validate_Foreign_Key_Checks/ Check_Customer_Name
SB-INVCIT-25313	Address_Id is invalid.	Validate_Foreign_Key_Checks/ Check_Address_Valid
SB-INVCIT-25314	Customer Category columns are invalid.	Validate_Foreign_Key_Checks/ Check_Cust_Cat_Code
SB-INVCIT-25315	Model customer item columns are invalid.	Validate_Foreign_Key_Checks/ Check_Model_Cust_Item_Id
SB-INVCIT-25316	Commodity code is inactive.	Perform_Other_Validations/ Check_Commodity_Active
SB-INVCIT-25317	Constainer_item_org_id is invalid.	Perform_Other_Validations/ Check_Container_Org_Id
SB-INVCIT-25318	Inactive flag has invalid value.	Perform_Other_Validations/ Check_Inactive_Flag
SB-INVCIT-25319	Container_org_code is invalid.	Perform_Other_Validations/ Check_Container_Org_Code
SB-INVCIT-25320	Container org name is invalid.	Perform_Other_Validations/ Check_Container_Org_Name
SB-INVCIT-25321	Demand tolerance positive value is invalid.	Perform_Other_Validations/ Check_Demand_Tolerance_Pstv
SB-INVCIT-25322	Demand tolerance negative value is invalid.	Perform_Other_Validations/ Check_Demand_Tolerance_Ngtv
SB-INVCIT-25323	Min_Fill_Percentage is invalid.	Perform_Other_Validations/ Check_Min_Fill_Percent
SB-INVCIT-25324	Departure plan required flag is invalid.	Perform_Other_Validations/ Check_Dep_Plan_Reqd_Flag
SB-INVCIT-25325	Departure plan build flag is invalid.	Perform_Other_Validations/ Check_Dep_Plan_Build_Flag
SB-INVCIT-25326	Departure plan flags combination is invalid.	Perform_Other_Validations/ Check_Dep_Plan_Flags

B.1.2 Customer Item Cross-Reference

Table 29 sb_validate_ci_xrefs_pkg

Error Code	Error Message	Generated from
SB-INVCCR-22601	Process_flag is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Process_flag
SB-INVCCR-22602	Process_mode is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Process_mode
SB-INVCCR-22603	Last_update_date is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Last_update_date
SB-INVCCR-22604	Last_updated_by is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Last_updated_by
SB-INVCCR-22605	Creation_date is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_creation_date
SB-INVCCR-22606	Created_by is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_created_by
SB-INVCCR-22607	Transaction_Type is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Type
SB-INVCCR-22608	Cust_Item_Number is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Cust_Item_Number
SB-INVCCR-22609	Preference_Number is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Preference_Number
SB-INVCCR-22610	Inactive_flag is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Inactive_flag
SB-INVCCR-22611	Lock_flag is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Lock_flag
SB-INVCCR-22612	Cust_Category is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Cust_Category
SB-INVCCR-22613	Address is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Address
SB-INVCCR-22614	Item_Def_Level is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Item_Def_Level
SB-INVCCR-22615	Customer is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Customer
SB-INVCCR-22616	Customer_Id is invalid.	Validate_Foreign_Keys/ Check_Customer_Id
SB-INVCCR-22617	Customer_No is invalid.	Validate_Foreign_Keys/ Check_Customer_No
SB-INVCCR-22618	Customer_Name is invalid.	Validate_Foreign_Keys/ Check_Customer_Name
SB-INVCCR-22619	Address_Valid is invalid.	Validate_Foreign_Keys/ Check_Address_Valid
SB-INVCCR-22620	Cust_Cat_Code is invalid.	Validate_Foreign_Keys/ Check_Cust_Cat_Code

Table 29 sb_validate_ci_xrefs_pkg

Error Code	Error Message	Generated from
SB-INVCCR-22621	Customer_item_id is invalid.	Validate_Foreign_Keys/ Check_Customer_item_id
SB-INVCCR-22622	Master_Organization_id is invalid.	Validate_Foreign_Keys/ Check_Master_Organization_id
SB-INVCCR-22623	Inventory_Item_id is invalid.	Validate_Foreign_Keys/ Check_Inventory_Item_id
SB-INVCCR-22624	Inactive_Flag is invalid.	Perform_Other_Validations/ Check_Inactive_Flag

B.1.3 Cycle Count Entries

Table 30 sb_validate_cc_entries_pkg

Error Code	Error Message	Generated from
SB-INVCCE-23301	CC_ENTRY_INTERFACE_ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_CC_Entry_Interface_ID
SB-INVCCE-23302	ORGANIZATION_ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Organization_ID
SB-INVCCE-23303	ACTION_CODE is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Action_Code
SB-INVCCE-23304	EMPLOYEE_ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Employee_ID
SB-INVCCE-23310	ADJUSTMENT_ACCOUNT_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Adjustment_Account_ID
SB-INVCCE-23311	CYCLE_COUNT_ENTRY_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Cycle_Count_Entry_ID
SB-INVCCE-23312	CYCLE_COUNT_HEADER_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Cycle_Count_Header_ID
SB-INVCCE-23313	LOCATOR_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Locator_ID
SB-INVCCE-23314	REVISION is invalid.	Validate_Foreign_Key_Checks/ Check_Revision
SB-INVCCE-23315	LOT_NUMBER is invalid.	Validate_Foreign_Key_Checks/ Check_Lot_Number
SB-INVCCE-23316	ORGANIZATION_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Organization_ID
SB-INVCCE-23317	SUBINVENTORY is invalid.	Validate_Foreign_Key_Checks/ Check_SubInventory
SB-INVCCE-23318	SERIAL_NUMBER is invalid.	Validate_Foreign_Key_Checks/ Check_Serial_Number
SB-INVCCE-23319	INVENTORY_ITEM_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Inventory_Item_ID
SB-INVCCE-23320	TRANSACTION_REASON_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Transaction_Reason_ID
SB-INVCCE-23321	COUNT_UNIT_OF_MEASURE is invalid.	Validate_Foreign_Key_Checks/ Check_Count_Unit_Of_Measure
SB-INVCCE-23322	COUNT_UOM is invalid.	Validate_Foreign_Key_Checks/ Check_Count_UOM
SB-INVCCE-23330	EMPLOYEE_ID value is not valid.	Perform_Other_Validations/ Check_EMPLOYEE_ID
SB-INVCCE-23331	PROJECT_ID value is not valid.	Perform_Other_Validations/ Check_Project_ID

Table 30 sb_validate_cc_entries_pkg

Error Code	Error Message	Generated from
SB-INVCCE-23332	TASK_ID value is not valid.	Perform_Other_Validations/ Check_Task_ID
SB-INVCCE-23333	VALID_FLAG value is not valid.	Perform_Other_Validations/ Check_Valid_Flag
SB-INVCCE-23334	DELETE_FLAG value is not valid.	Perform_Other_Validations/ Check_Delete_Flag
SB-INVCCE-23335	LOCK_FLAG value is not valid.	Perform_Other_Validations/ Check_Lock_Flag
SB-INVCCE-23336	PROCESS_FLAG value is not valid.	Perform_Other_Validations/ Check_Process_Flag
SB-INVCCE-23337	PROCESS_MODE value is not valid.	Perform_Other_Validations/ Check_Process_Mode
SB-INVCCE-23338	STATUS_FLAG value is not valid.	Perform_Other_Validations/ Check_Status_Flag

B.1.4 Items

Table 31 sb_validate_items_pkg

Error Code	Error Message	Generated from
SB-INVITM-21301	Organization_Code or Organization_Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Organization_Id
SB-INVITM-21302	item_Number or Segment#s 1 to 20 is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Item
SB-INVITM-21303	Process_Flag is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Process_Flag
SB-INVITM-21304	Transaction_Type is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Tran_Type
SB-INVITM-21305	Set_Process_Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Proc_Id
SB-INVITM-21306	Summary_Flag is invalid.	Perform_Other_Validations/ Check_Summary_Flag
SB-INVITM-21307	Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Enabled_Flag
SB-INVITM-21308	Purchasing_Item_Flag is invalid.	Perform_Other_Validations/ Check_Purchase_Item_Flag
SB-INVITM-21309	Shippable_Item_Flag is invalid.	Perform_Other_Validations/ Check_Ship_Item_Flag
SB-INVITM-21310	Customer_Order_Flag is invalid.	Perform_Other_Validations/ Check_Cust_Order_Flag
SB-INVITM-21311	Internal_Order_Flag is invalid.	Perform_Other_Validations/ Check_Int_Order_Flag
SB-INVITM-21312	Service_Item_Flag is invalid.	Perform_Other_Validations/ Check_Service_Item_Flag
SB-INVITM-21313	Service_Starting_Delay is less than Zero.	Perform_Other_Validations/ Check_Service_Start_Delay
SB-INVITM-21314	Inventory_Item_Flag is invalid.	Perform_Other_Validations/ Check_Inv_Item_Flag
SB-INVITM-21315	Eng_Item_Flag is invalid.	Perform_Other_Validations/ Check_Eng_Item_Flag
SB-INVITM-21316	Inventory_Asset_Flag is invalid.	Perform_Other_Validations/ Check_Inv_Asset_Flag
SB-INVITM-21317	Purchasing_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Pur_Enabled_Flag
SB-INVITM-21318	Customer_Order_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Cust_Order_Enabled_Flag
SB-INVITM-21319	Internal_Order_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Int_Order_Enabled_Flag

Table 31 sb_validate_items_pkg

Error Code	Error Message	Generated from
SB-INVITM-21320	So_Transactions_Flag is invalid.	Perform_Other_Validations/ Check_SO_Trans_Flag
SB-INVITM-21321	Mtl_Transactions_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_MTL_Trans_Enabled_Flag
SB-INVITM-21322	Stock_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Stock_Enabled_Flag
SB-INVITM-21323	Bom_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_BOM_Enabled_Flag
SB-INVITM-21324	Build_In_Wip_Flag is invalid.	Perform_Other_Validations/ Check_Buildin_WIP_Flag
SB-INVITM-21325	Allow_Item_Desc_Update_Flag is invalid.	Perform_Other_Validations/ Check_Item_Desc_Upd_Flag
SB-INVITM-21326	Receipt_Required_Flag is invalid.	Perform_Other_Validations/ Check_Receipt_Required_Flag
SB-INVITM-21327	Rfq_Required_Flag is invalid.	Perform_Other_Validations/ Check_RFQ_Reqd_Flag
SB-INVITM-21328	Planning_Time_Fence_Days is less than Zero.	Perform_Other_Validations/ Check_Plan_Time_Fence_Days
SB-INVITM-21329	Pick_Components_Flag is invalid.	Perform_Other_Validations/ Check_Pick_Comps_Flag
SB-INVITM-21330	Replenish_To_Order_Flag is invalid.	Perform_Other_Validations/ Check_Repl_To_Order_Flag
SB-INVITM-21331	ATP_Components_Flag is invalid.	Perform_Other_Validations/ Check_ATP_Components_Flag
SB-INVITM-21332	Cost_Of_Sales_Account is invalid.	Validate_Foreign_Key_Checks/ Check_Cost_of_Sales_Act
SB-INVITM-21333	Sales_Account is invalid.	Validate_Foreign_Key_Checks/ Check_Sales_Act_DSP
SB-INVITM-21334	Encumbrance_Account is invalid.	Validate_Foreign_Key_Checks/ Check_Encumbrance_Account
SB-INVITM-21335	Expense_Account is invalid.	Validate_Foreign_Key_Checks/ Check_Expense_Account
SB-INVITM-21336	List_Price_Per_Unit is less than Zero.	Perform_Other_Validations/ Check_List_Price_Unit
SB-INVITM-21337	Taxable_Flag is invalid.	Perform_Other_Validations/ Check_Taxable_Flag
SB-INVITM-21338	Expense_Billable_Flag is invalid.	Perform_Other_Validations/ Check_Expense_Billable_Flag
SB-INVITM-21339	Time_Billable_Flag is invalid.	Perform_Other_Validations/ Check_Time_Billable_Flag
SB-INVITM-21340	Service_Duration is invalid.	Perform_Other_Validations/ Check_Service_Duration

Table 31 sb_validate_items_pkg

Error Code	Error Message	Generated from
SB-INVITM-21341	Market_Price is less than Zero.	Perform_Other_Validations/ Check_Market_Price
SB-INVITM-21342	Price_Tolerance_Percent is less than Zero.	Perform_Other_Validations/ Check_Price_Tolerance_pc
SB-INVITM-21343	Shelf_Life_Days is less than Zero.	Perform_Other_Validations/ Check_Shelf_Life
SB-INVITM-21344	Repetitive_Planning_Flag is invalid.	Perform_Other_Validations/ Check_Rep_Plan_Flag
SB-INVITM-21345	Acceptable_Rate_Decrease is less than Zero.	Perform_Other_Validations/ Check_Rate_Decrease
SB-INVITM-21346	Acceptable_Rate_Increase is less than Zero.	Perform_Other_Validations/ Check_Rate_Increase
SB-INVITM-21347	Assembly_Pegging_Flag is invalid.	Perform_Other_Validations/ Check_Assembly_Pegging_Flag
SB-INVITM-21348	Postprocessing_Lead_Time is less than Zero.	Perform_Other_Validations/ Check_Post_Proc_Lead_Time
SB-INVITM-21349	Vendor_Warranty_Flag is invalid.	Perform_Other_Validations/ Check_Vendor_Warranty_Flag
SB-INVITM-21350	Serviceable_Component_Flag is invalid.	Perform_Other_Validations/ Check_Serv_Comp_Flag
SB-INVITM-21351	Serviceable_Product_Flag is invalid.	Perform_Other_Validations/ Check_Serv_Product_Flag
SB-INVITM-21352	Preventive_Maintenance_Flag is invalid.	Perform_Other_Validations/ Check_Prevent_Maint_Flag
SB-INVITM-21353	Ship_Model_Complete_Flag is invalid.	Perform_Other_Validations/ Check_Ship_Model_Complete_Flag
SB-INVITM-21354	Prorate_Service_Flag is invalid.	Perform_Other_Validations/ Check_Prorate_Service_Flag
SB-INVITM-21355	Invoiceable_Item_Flag is invalid.	Perform_Other_Validations/ Check_Invoice_Item_Flag
SB-INVITM-21356	Invoice_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Invoice_Enabled_Flag
SB-INVITM-21357	Must_Use_Approved_Vendor_Flag is invalid.	Perform_Other_Validations/ Check_MustUse_ApprVendor_Flag
SB-INVITM-21358	Outside_Operation_Flag is invalid.	Perform_Other_Validations/ Check_Outside_Operation_Flag
SB-INVITM-21359	Costing_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Costing_Enabled_Flag
SB-INVITM-21360	Cycle_Count_Enabled_Flag is invalid.	Perform_Other_Validations/ Check_Cycle_Count_Enabled_Flag
SB-INVITM-21361	Auto_Created_Config_Flag is invalid.	Perform_Other_Validations/ Check_Auto_Created_Config_Flag

Table 31 sb_validate_items_pkg

Error Code	Error Message	Generated from
SB-INVITM-21362	Container_Item_Flag is invalid.	Perform_Other_Validations/ Check_Container_Item_Flag
SB-INVITM-21363	Vehicle_Item_Flag is invalid.	Perform_Other_Validations/ Check_Vehicle_Item_Flag
SB-INVITM-21364	Check_Org_ID_Code is invalid.	Perform_Other_Validations/ Check_Org_ID_Code
SB-INVITM-21365	Wip_Supply_Type is invalid.	Perform_Other_Validations/ Check_WIP_Supply_Type
SB-INVITM-21366	Lot_Control_Code is invalid.	Perform_Other_Validations/ Check_Lot_Control_Code
SB-INVITM-21367	Shelf_Life_Code is invalid.	Perform_Other_Validations/ Check_Shelf_Life_Code
SB-INVITM-21368	Serial_Number_Control_Code is invalid.	Perform_Other_Validations/ Check_SI_No_Control_Code
SB-INVITM-21369	Restrict_Subinventories_Code is invalid.	Perform_Other_Validations/ Check_Sub_Inv_Code
SB-INVITM-21370	Restrict_Locators_Code is invalid.	Perform_Other_Validations/ Check_Restrict_Locators_Code
SB-INVITM-21371	Planning_Time_Fence_Code is invalid.	Perform_Other_Validations/ Check_Plan_Time_Fence_Code
SB-INVITM-21372	Bom_Item_Type is invalid.	Perform_Other_Validations/ Check_BOM_Item_Type
SB-INVITM-21373	Allowed_Units_Lookup_Code is invalid.	Perform_Other_Validations/ Check_Allowed_Units_Code
SB-INVITM-21374	Inventory_Planning_Code is invalid.	Perform_Other_Validations/ Check_Inv_Planning_Code
SB-INVITM-21375	Planning_Make_Buy_Code is invalid.	Perform_Other_Validations/ Check_Plan_Make_Buy_Code
SB-INVITM-21376	Mrp_Safety_Stock_Code is invalid.	Perform_Other_Validations/ Check_MRP_Safety_stock_Code
SB-INVITM-21377	Material_Billable_Flag is invalid.	Perform_Other_Validations/ Check_Material_Billable_Flag
SB-INVITM-21378	Reservable_Type is invalid.	Perform_Other_Validations/ Check_Reservable_Type
SB-INVITM-21379	Return_Inspection_Requirement is invalid.	Perform_Other_Validations/ Check_Return_Insp_Reqmt
SB-INVITM-21380	Mrp_Planning_Code is invalid.	Perform_Other_Validations/ Check_MRP_Planning_Code
SB-INVITM-21381	Location_Control_Code is invalid.	Perform_Other_Validations/ Check_Location_Control_Code
SB-INVITM-21382	ITEM DESCRIPTION FIELD CANNOT BE NULL.	Validate_Required_Columns/ Check_Item_Desc

B.1.5 Item Revisions

Table 32 sb_validate_invrev_pkg

Error Code	Error Message	Generated from
SB-INVREV-22500	Set_process_ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Set_process_ID
SB-INVREV-22501	Inventory_Item_Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Inventory_Item_Id
SB-INVREV-22502	Organization_Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Organization_Id
SB-INVREV-22503	Revision is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Revision
SB-INVREV-22504	Change_Notice is invalid.	Validate_Foreign_Key_Checks/ Check_Change_Notice
SB-INVREV-22505	Organization_id is invalid.	Validate_Foreign_Key_Checks/ Check_Organization_id
SB-INVREV-22506	Inventory_item_id is invalid.	Validate_Foreign_Key_Checks/ Check_Inventory_item_id
SB-INVREV-22507	Primary_key is invalid.	Validate_Foreign_Key_Checks/ Check_Primary_key
SB-INVREV-22508	Process_flag is invalid.	Perform_Other_Validations/ Check_Process_flag
SB-INVREV-22509	Item_Number is invalid.	Perform_Other_Validations/ Check_Item_Number

B.1.6 Replenishment

Master

Table 33 sb_validate_rep_pkg

Error Code	Error Message	Generated from
SB-INVREP-22430	Header has lines that have failed validation.	Validate_REC/Validate

Headers

Table 34 sb_validate_rep_hdrs_pkg

Error Code	Error Message	Generated from
SB-INVREP-22301	Replenishment_Header_ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Replenishment_Header_ID
SB-INVREP-22302	Replenishment_Count_Name is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Replenishment_Count_Name
SB-INVREP-22303	Count_Date is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Count_Date
SB-INVREP-22304	Last_Update_Date is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Last_Update_Date
SB-INVREP-22305	Creation_Date is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Creation_Date
SB-INVREP-22306	Created_By is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Created_By
SB-INVREP-22307	Last_Updated_By is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Last_Updated_By
SB-INVREP-22308	Organization_ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Organization_ID
SB-INVREP-22309	SubInventory_Code is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_SubInventory_Code
SB-INVREP-22310	Process_Status is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Process_Status
SB-INVREP-22311	Process_Mode is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Process_Mode
SB-INVREP-22312	Organization_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Organization_ID
SB-INVREP-22313	SubInventory_Code is invalid.	Validate_Foreign_Key_Checks/ Check_SubInventory_Code
SB-INVREP-22314	Error_Flag is invalid.	Perform_Other_Validations/ Check_Error_Flag

Table 34 sb_validate_rep_hdrs_pkg

Error Code	Error Message	Generated from
SB-INVREP-22315	Process_Mode is invalid.	Perform_Other_Validations/ Check_Process_Mode
SB-INVREP-22316	Process_Status is invalid.	Perform_Other_Validations/ Check_Process_Status
SB-INVREP-22317	Replenish_Header is invalid.	Perform_Other_Validations/ Check_Replenish_Header
SB-INVREP-22318	Replenish_Count is invalid.	Perform_Other_Validations/ Check_Replenish_Count
SB-INVREP-22319	Last_Update is invalid.	Perform_Other_Validations/ Check_Last_Update
SB-INVREP-22320	Lines is invalid.	Perform_Other_Validations/ Check_Lines

Lines

Table 35 sb_validate_rep_line_pkg

Error Code	Error Message	Generated from
SB-INVREP-22401	Replenishment_Header_ID is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Replenishment_Header_ID
SB-INVREP-22402	Replenishment_Line_Id is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Replenishment_Line_Id
SB-INVREP-22403	Last_Update_Date is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Last_Update_Date
SB-INVREP-22404	Creation_Date is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Creation_Date
SB-INVREP-22405	Created_By is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Created_By
SB-INVREP-22406	Last_Update_Login is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Last_Update_Login
SB-INVREP-22407	Last_Updated_By is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Last_Updated_By
SB-INVREP-22408	Inventory_Item_ID is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Inventory_Item_ID
SB-INVREP-22409	Count_Type_Code is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Count_Type_Code
SB-INVREP-22410	Count_Quantity is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Count_Quantity
SB-INVREP-22411	Count_UOM is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Count_UOM
SB-INVREP-22412	Count_UOM_Code is mandatory and the current value is NULL	Validate_Required_Columns/ Check_Count_UOM_Code

Table 35 sb_validate_rep_line_pkg

Error Code	Error Message	Generated from
SB-INVREP-22413	Organization_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Organization_ID
SB-INVREP-22414	Rep_Header_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Rep_Header_ID
SB-INVREP-22415	Inventory_Item_ID is invalid.	Validate_Foreign_Key_Checks/ Check_Inventory_Item_ID
SB-INVREP-22416	Count_Uom is invalid.	Validate_Foreign_Key_Checks/ Check_Count_Uom
SB-INVREP-22417	Count_Uom_Code is invalid.	Validate_Foreign_Key_Checks/ Check_Count_Uom_Code
SB-INVREP-22418	Segment is invalid.	Validate_Foreign_Key_Checks/ Check_Segment
SB-INVREP-22419	Count_Type_Code is invalid.	Perform_Other_Validations/ Check_Count_Type_Code
SB-INVREP-22420	Organization_Id is invalid.	Perform_Other_Validations/ Check_Organization_Id
SB-INVREP-22421	Count_Type_Tracked is invalid.	Perform_Other_Validations/ Check_Count_Type_Tracked
SB-INVREP-22421	Count_Type_Tracked23 is invalid.	Perform_Other_Validations/ Check_Count_Type_Tracked23
SB-INVREP-22422	Inventory_item is invalid.	Perform_Other_Validations/ Check_Inventory_item
SB-INVREP-22423	MinMAx_Plan is invalid.	Perform_Other_Validations/ Check_MinMAx_Plan
SB-INVREP-22424	Count_uom_primary is invalid.	Perform_Other_Validations/ Check_Count_uom_primary

B.1.7 Transactions

Table 36 sb_validate_trans_pkg

Error Code	Error Message	Generated from
SB-INVTXN-20301	Source Code is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Source_Code
SB-INVTXN-20302	Source Line Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Source_Line_ID
SB-INVTXN-20303	Source Header Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Source_Header_ID
SB-INVTXN-20304	Process Flag is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Process_Flag
SB-INVTXN-20305	Transaction Mode is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Mode
SB-INVTXN-20306	Transaction Interface Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Interface_ID
SB-INVTXN-20307	Inventory Item Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Inventory_Item_ID
SB-INVTXN-20308	Organization Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Organization_ID
SB-INVTXN-20309	Transaction Quantity is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Quantity
SB-INVTXN-20310	Transaction UOM is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_UOM
SB-INVTXN-20311	Transaction Date is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Date
SB-INVTXN-20312	Transaction Source Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Source_ID
SB-INVTXN-20313	Transaction Source Name is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Source_Name
SB-INVTXN-20314	Transaction Type Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Transaction_Type_ID
SB-INVTXN-20319	Distribution Account Id is invalid.	Validate_Foreign_Key_Checks/ Check_Distribution_Account_ID
SB-INVTXN-20320	Inventory Item Id is invalid.	Validate_Foreign_Key_Checks/ Check_Inventory_Item_ID
SB-INVTXN-20321	Organization Id is invalid.	Validate_Foreign_Key_Checks/ Check_Organization_ID
SB-INVTXN-20322	Acct Period Id is invalid.	Validate_Foreign_Key_Checks/ Check_Acct_Period_ID
SB-INVTXN-20323	Schedule Id is invalid.	Validate_Foreign_Key_Checks/ Check_Schedule_ID
SB-INVTXN-20324	Operation Seq Num is invalid.	Validate_Foreign_Key_Checks/ Check_Operation_Seq_Num

Table 36 sb_validate_trans_pkg

Error Code	Error Message	Generated from
SB-INVTXN-20325	Transportation Account is invalid.	Validate_Foreign_Key_Checks/ Check_Transportation_Account
SB-INVTXN-20326	Locator Id is invalid.	Validate_Foreign_Key_Checks/ Check_Locator_ID
SB-INVTXN-20327	Transfer Organization is invalid.	Validate_Foreign_Key_Checks/ Check_Transfer_Organization
SB-INVTXN-20328	Transfer Locator is invalid.	Validate_Foreign_Key_Checks/ Check_Transfer_Locator
SB-INVTXN-20329	Revision is invalid.	Validate_Foreign_Key_Checks/ Check_Revision
SB-INVTXN-20330	SubInventory Code is invalid.	Validate_Foreign_Key_Checks/ Check_Sub_inventory_Code
SB-INVTXN-20331	Transfer SubInventory is invalid.	Validate_Foreign_Key_Checks/ Check_Transfer_SubInventory
SB-INVTXN-20332	Schedule Number is invalid.	Validate_Foreign_Key_Checks/ Check_Schedule_Number
SB-INVTXN-20333	Reason Id is invalid.	Perform_Other_Validations/ Check_Reason_ID
SB-INVTXN-20334	Wip Entity Type is invalid.	Perform_Other_Validations/ Check_WIP_Entity_Type
SB-INVTXN-20335	Substitution Type Id is invalid.	Perform_Other_Validations/ Check_Substitution_ID
SB-INVTXN-20344	Transaction Header Id is invalid.	Perform_Other_Validations/ Check_Trans_Header_ID

Lots

Table 37 sb_validate_trans_lots_pkg

Error Code	Error Message	Generated from
SB-INVTXN-20336	Transaction Interface Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Tran_Interface_ID
SB-INVTXN-20337	Lot Number is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Lot_Number
SB-INVTXN-20338	Transaction Quantity is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Tran_Quantity
SB-INVTXN-20339	Transaction Temp Id is invalid.	Validate_Foreign_Key_Checks/ Check_Serial_Tran_Temp_ID
SB-INVTXN-20340	Transaction Interface Id is invalid.	Validate_Foreign_Key_Checks/ Check_Tran_Interface_ID

Serial Numbers

Table 38 sb_validate_serial_nos_pkg

Error Code	Error Message	Generated from
SB-INVTXN-20341	Transaction Interface Id is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Tran_Interface_ID
SB-INVTXN-20342	Fm Serial Number is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Fm_Serial_Number
SB-INVTXN-20343	Vendor Serial Number is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_Serial_No

B.2 Purchase Order

B.2.1 Import

Master

Table 39 sb_validate_poi_pkg

Error Code	Error Message	Generated from
SB-POIMP-20001	Header record does not exist for Po_Header_Id.	Validate_Master_Details
SB-POIMP-25006	Header has lines that have failed validation.	Validate

Headers

Table 40 sb_validate_poi_hdrs_pkg

Error Code	Error Message	Generated from
SB-POIMP-20002	Interface Header ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Interface_Header_ID
SB-POIMP-20003	Action is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Action
SB-POIMP-20004	Document Type Code is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Document_Type_Code
SB-POIMP-20006	Vendor Site Code is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Vendor_Site_Code
SB-POIMP-20007	Vendor Doc Num is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Vendor_Doc_Num
SB-POIMP-20008	Action is mandatory and the current value is invalid.	Validate_Required_Columns/ Check_Action_Values
SB-POIMP-20009	Document Type Code is mandatory and the current value is invalid.	Validate_Required_Columns/ Check_Doc_Type_Code_Values
SB-POIMP-20010	Vendor ID is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_ID
SB-POIMP-20011	From Type Lookup Code is invalid.	Validate_Foreign_Key_Checks/ Check_From_Type_Lookup_Code
SB-POIMP-20012	Ship To Location Id is invalid.	Validate_Foreign_Key_Checks/ Check_Ship_To_Location_ID
SB-POIMP-20013	Terms Id is invalid.	Validate_Foreign_Key_Checks/ Check_Terms_ID

Table 40 sb_validate_poi_hdrs_pkg

Error Code	Error Message	Generated from
SB-POIMP-20014	Ussgl Transaction Code is invalid.	Validate_Foreign_Key_Checks/ Check_Ussgl_Transaction_Code
SB-POIMP-20015	Buyer specified is invalid.	Validate_Foreign_Key_Checks/ Check_Agent_ID
SB-POIMP-20016	Bill To Location Id is invalid.	Validate_Foreign_Key_Checks/ Check_Bill_To_Location_ID
SB-POIMP-20017	Vendor Site Id is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_Site_ID
SB-POIMP-20018	Vendor Contact Id is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_Contact_ID
SB-POIMP-20019	From Header Id is invalid.	Validate_Foreign_Key_Checks/ Check_From_Header_ID
SB-POIMP-20020	Rate Type is invalid.	Validate_Foreign_Key_Checks/ Check_Rate_Type
SB-POIMP-20021	Currency Code is invalid.	Validate_Foreign_Key_Checks/ Check_Currency_Code
SB-POIMP-20022	Document Number is invalid.	Validate_Foreign_Key_Checks/ Check_Document_Num
SB-POIMP-20023	Buyer Name is invalid.	Validate_Foreign_Key_Checks/ Check_Agent_Name
SB-POIMP-20024	Vendor Name or Vendor Num is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_Name
SB-POIMP-20025	Vendor Site Code is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_Site_Code
SB-POIMP-20026	Vendor Contact is invalid.	Validate_Foreign_Key_Checks/ Check_Vendor_Contact
SB-POIMP-20027	Ship To Location Code is invalid.	Validate_Foreign_Key_Checks/ Check_Ship_To_Location_Code
SB-POIMP-20028	Bill To Location Code is invalid.	Validate_Foreign_Key_Checks/ Check_Bill_To_Location_Code
SB-POIMP-20029	Payment Terms is invalid.	Validate_Foreign_Key_Checks/ Check_Payment_Terms
SB-POIMP-20030	Rfq Num is invalid.	Validate_Foreign_Key_Checks/ Check_Rfq_Num
SB-POIMP-20031	Amount Limit is less than Amount Agreed.	Perform_Other_Validations/ Check_Amount_Agreed
SB-POIMP-20032	Amount Limit is less than Min Release Amount.	Perform_Other_Validations/ Check_Min_Release_Amount
SB-POIMP-20033	Amount Limit is less than Zero.	Perform_Other_Validations/ Check_Amount_Limit
SB-POIMP-20088	Release Number is invalid.	Perform_Other_Validations/ Check_Release_Num

Table 40 sb_validate_poi_hdrs_pkg

Error Code	Error Message	Generated from
SB-POIMP-25007	Document Type is Standard, so Action cannot be Update or Replace.	Perform_Other_Validations/ Check_Action_For_Std_Doc_Type
SB-POIMP-25008	Document Type is Standard, so Approval Status should be Incomplete.	Perform_Other_Validations/ Chk_App_Stat_For_Std_Doc_Type
SB-POIMP-25009	Vendor Doc Num does not exist.	Perform_Other_Validations/ <ul style="list-style-type: none"> ▪ Check_Vendor_Doc_Num_Repl_Quot ▪ Check_Vendor_Doc_Num_Repl_Blkt ▪ Check_Vendor_Doc_Num_Upd_Quot ▪ Check_Vendor_Doc_Num_Upd_Blkt
SB-POIMP-25010	Vendor Doc Num already exists.	Perform_Other_Validations/ <ul style="list-style-type: none"> ▪ Check_Vendor_Doc_Num_Org_Quot ▪ Check_Vendor_Doc_Num_Org_Blkt
SB-POIMP-25011	Release Date is reserved for future use and should be NULL.	Perform_Other_Validations/ Check_Release_Date
SB-POIMP-25012	Document Num must have a unique value, Current value already exists.	Validate_Foreign_Key_Checks/ Check_Document_Num_For_Orig
SB-POIMP-25013	There should be at least one line per header document.	Perform_Other_Validations/ Check_Lines_For_Header
SB-POIMP-25014	Fob_Active is invalid.	Validate_Foreign_Key_Checks/ Check_Fob_Active
SB-POIMP-25015	Freight_Terms_Active is invalid.	Validate_Foreign_Key_Checks/ Check_Freight_Terms_Active
SB-POIMP-25016	Document_Sub_Type is invalid.	Perform_Other_Validations/ Check_Document_Sub_Type
SB-POIMP-25017	Num_Of_Lines is invalid.	Perform_Other_Validations/ Check_Num_Of_Lines

Lines

Table 41 sb_validate_poi_lines_pkg

Error Code	Error Message	Generated from
SB-POIMP-20034	Interface Header ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Interface_Header_ID
SB-POIMP-20035	Interface Line ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Interface_Line_ID
SB-POIMP-20036	Organization ID is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Organization_ID
SB-POIMP-20037	Po Header Id not found for modification.	Validate_Foreign_Key_Checks/ Check_Po_Header_ID
SB-POIMP-20038	Category Id is invalid.	Validate_Foreign_Key_Checks/ Check_Category_ID
SB-POIMP-20039	Line Type Id is invalid.	Validate_Foreign_Key_Checks/ Check_Line_Type_ID
SB-POIMP-20040	Un Number Id is invalid.	Validate_Foreign_Key_Checks/ Check_Un_Number_ID
SB-POIMP-20041	Hazard Class Id is invalid.	Validate_Foreign_Key_Checks/ Check_Hazard_Class_ID
SB-POIMP-20042	Ussgl Transaction Code is invalid.	Validate_Foreign_Key_Checks/ Check_Ussgl_Transaction_Code
SB-POIMP-20043	Closed By is invalid.	Validate_Foreign_Key_Checks/ Check_Closed_By
SB-POIMP-20044	From Header Id is invalid.	Validate_Foreign_Key_Checks/ Check_From_Header_ID
SB-POIMP-20045	From Line Id is invalid.	Validate_Foreign_Key_Checks/ Check_From_Line_ID
SB-POIMP-20046	Tax Name is invalid.	Validate_Foreign_Key_Checks/ Check_Tax_Name
SB-POIMP-20047	Item Revision is invalid.	Validate_Foreign_Key_Checks/ Check_Item_Revision
SB-POIMP-20048	Unit Of Measure is invalid.	Validate_Foreign_Key_Checks/ Check_Unit_Of_Measure
SB-POIMP-20049	Unit Of Measure is invalid for the Item ID/Organization ID.	Validate_Foreign_Key_Checks/ Check_Unit_Of_Measure_Org
SB-POIMP-20050	Item Id is invalid.	Validate_Foreign_Key_Checks/ Check_Item_Id
SB-POIMP-20052	Item is invalid.	Validate_Foreign_Key_Checks/ Check_Item
SB-POIMP-20054	Uom Code is invalid.	Validate_Foreign_Key_Checks/ Check_Uom_Code
SB-POIMP-20055	Line Type is invalid.	Validate_Foreign_Key_Checks/ Check_Line_Type

Table 41 sb_validate_poi_lines_pkg

Error Code	Error Message	Generated from
SB-POIMP-20056	Line Num is invalid.	Validate_Foreign_Key_Checks/ Check_Line_Num
SB-POIMP-20057	Shipment Num is invalid.	Validate_Foreign_Key_Checks/ Check_Shipment_Num
SB-POIMP-20058	Un Number is invalid.	Validate_Foreign_Key_Checks/ Check_Un_Number
SB-POIMP-20059	Hazard Class is invalid.	Validate_Foreign_Key_Checks/ Check_Hazard_Class
SB-POIMP-20060	Ship To Organization Code is invalid.	Validate_Foreign_Key_Checks/ Check_Ship_To_Org_Code
SB-POIMP-20061	Ship To Location Code is invalid.	Validate_Foreign_Key_Checks/ Check_Ship_To_Location_Code
SB-POIMP-20062	Template Name is invalid.	Validate_Foreign_Key_Checks/ Check_Template_Name
SB-POIMP-20063	Payment Terms is invalid.	Validate_Foreign_Key_Checks/ Check_Payment_Terms
SB-POIMP-20064	Receiving Routing is invalid.	Validate_Foreign_Key_Checks/ Check_Receiving_Routing
SB-POIMP-20065	Tax Code Id is invalid.	Validate_Foreign_Key_Checks/ Check_Tax_Code_Id
SB-POIMP-20066	Not To Exceed Price should be NULL since Allow Price Override Flag is N.	Perform_Other_Validations/ Check_Not_To_Exceed_Price
SB-POIMP-20067	Not To Exceed Price has to be greater or equal to Unit Price.	Perform_Other_Validations/ Check_Unit_Price
SB-POIMP-20068	Max Order Quantity should be greater than Min Order Quantity.	Perform_Other_Validations/ Check_Order_Quantity
SB-POIMP-20069	Closed Code should be null for QUOTATION Document Type.	Perform_Other_Validations/ Check_Closed_Code_For_Quot
SB-POIMP-20070	Committed Amount should be null for QUOTATION Document Type.	Perform_Other_Validations/ Check_Cmtd_Amt_For_Quot
SB-POIMP-20071	Market Price should be null for QUOTATION Document Type.	Perform_Other_Validations/ Check_Market_Price_For_Quot
SB-POIMP-20072	Allow Price Override Flag should be null for QUOTATION Document Type.	Perform_Other_Validations/ Chk_Alw_Prc_Ovd_Flag_For_Quot
SB-POIMP-20073	Not To Exceed Price should be null for QUOTATION Document Type.	Perform_Other_Validations/ Check_Not_To_Exc_Prc_For_Quot
SB-POIMP-20074	Negotiated By Preparer Flag should be null for QUOTATION Document Type.	Perform_Other_Validations/ Chk_Neg_By_Prep_Flag_For_Quot
SB-POIMP-20075	Capital Expense Flag should be null for QUOTATION Document Type.	Perform_Other_Validations/ Check_Cap_Exp_Flag_For_Quot
SB-POIMP-20076	Min Release Amount for QUOTATION/ BLANKET Document Type.	Perform_Other_Validations/ Check_Min_Rel_Amt

Table 41 sb_validate_poi_lines_pkg

Error Code	Error Message	Generated from
SB-POIMP-20077	Min Order Quantity is less than 0.	Perform_Other_Validations/ Check_Min_Ord_Qty_For_Quot
SB-POIMP-20078	Max Order Quantity is less than 0.	Perform_Other_Validations/ Check_Max_Ord_Qty_For_Quot
SB-POIMP-20079	Over Tolerance Error Flag should be either Y or N.	Perform_Other_Validations/ Check_Ovr_Tol_Error_Flg
SB-POIMP-20080	Qty Rcv Tolerance should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Chk_Qty_Rcv_Tol_For_Blkt
SB-POIMP-20081	Qty Rcv Exception Code should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Chk_Qty_Rcv_Exc_Code_For_Blkt
SB-POIMP-20082	Freight Carrier should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Chk_Freight_Carrier_For_Blkt
SB-POIMP-20083	Fob should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Check_Fob_For_Blkt
SB-POIMP-20084	Freight Terms should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Check_Freight_Terms_For_Blkt
SB-POIMP-20085	Receipt Required Flag should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Check_Rcpt_Req_Flag_For_Blkt
SB-POIMP-20086	Inspection Required Flag should be NULL for BLANKET Document Type.	Perform_Other_Validations/ Check_Insp_Req_Flag_For_Blkt
SB-POIMP-20087	Unit Price should not be NULL or < 0.	Perform_Other_Validations/ Check_Unit_Price_For_Null
SB-POIMP-20088	Release Num is reserved for future use and should be NULL.	Perform_Other_Validations/ Check_Release_Num
SB-POIMP-20089	Po Release Id is reserved for future use and should be NULL.	Perform_Other_Validations/ Check_Po_Release_Id
SB-POIMP-20090	Source Shipment Id is reserved for future use and should be NULL.	Perform_Other_Validations/ Check_Source_Shipment_Id
SB-POIMP-20091	Contract Num is reserved for future use and should be NULL.	Perform_Other_Validations/ Check_Contract_Num
SB-POIMP-20092	Type 1099 should be NULL.	Perform_Other_Validations/ Check_Type_1099
SB-POIMP-20093	Receipt Days Exception Code should be NULL.	Perform_Other_Validations/ Check_Rcpt_Days_Exp_Cd
SB-POIMP-20094	Need By Date should be NULL.	Perform_Other_Validations/ Check_Need_By_Date
SB-POIMP-20095	Promised Date should be NULL.	Perform_Other_Validations/ Check_Promised_Date
SB-POIMP-20096	Closed Reason should be NULL.	Perform_Other_Validations/ Check_Closed_Reason
SB-POIMP-20097	Closed Date should be NULL.	Perform_Other_Validations/ Check_Closed_Date

Table 41 sb_validate_poi_lines_pkg

Error Code	Error Message	Generated from
SB-POIMP-20098	Closed By should be NULL.	Perform_Other_Validations/ Check_Closed_By
SB-POIMP-20099	Firm Flag should be NULL.	Perform_Other_Validations/ Check_Firm_Flag
SB-POIMP-20100	Quantity should be greater than 0.	Perform_Other_Validations/ Check_Po_Release_Id
SB-POIMP-25001	Committed Amount should be greater than 0.	Perform_Other_Validations/ Check_Committed_Amount
SB-POIMP-25002	Market Price should be greater than 0.	Perform_Other_Validations/ Check_Market_Price
SB-POIMP-25003	Negotiated By Preparer Flag should be either Y or N.	Perform_Other_Validations/ Check_Neg_By_Prep_Flg
SB-POIMP-25004	Capital Expense Flag should be either Y or N.	Perform_Other_Validations/ Check_Capital_Exp_Flg
SB-POIMP-25005	Tax name and Tax Code Id should be NULL if Taxable Flag is NULL or N.	Perform_Other_Validations/ Check_Taxable_Flag
SB-POIMP-25014	FOB is inactive.	Validate_Foreign_Key_Checks/ Check_Fob_Active
SB-POIMP-25015	Freight_Terms is inactive.	Validate_Foreign_Key_Checks/ Check_Freight_Terms_Active

B.2.2 Receiving

Master

Table 42 sb_validate_rcv_pkg

Error Code	Error Message	Generated from
SB-POREC-25006	Header has lines that have failed validation.	Validate

Header

Table 43 sb_validate_rcv_hdr_pkg

Error Code	Error Message	Generated from
SB-POREC-20100	Header_Interface_ID field is mandatory in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_header_interface_id
SB-POREC-20101	Group_ID field is mandatory in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_group_id
SB-POREC-20102	Processing_Status_Code field is mandatory in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_processing_status_code
SB-POREC-20103	Receipt_Source_Code field is mandatory in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_receipt_source_code
SB-POREC-20104	Transaction_Type field is mandatory in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_transaction_type
SB-POREC-20105	Value Required in Vendor_Name or Vendor_Num field in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_vendor_name_num
SB-POREC-20106	Validation_Flag is mandatory in RCV_HEADERS_INTERFACE table and current value is null.	Validate_Required_Columns/ Check_validation_flag
SB-POREC-20107	ASN_TYPE field in RCV_HEADERS_INTERFACE should be ASN or ASBN.	Derive_Required_Columns/ Check_asn_type
SB-POREC-20108	Processing_Status_Code field in RCV_HEADERS_INTERFACE should be PENDING.	Derive_Required_Columns/ Check_processing_status_code
SB-POREC-20109	Receipt_Source_Code field in RCV_HEADERS_INTERFACE should be VENDOR.	Derive_Required_Columns/ Check_receipt_source_code

Table 43 sb_validate_rcv_hdr_pkg

Error Code	Error Message	Generated from
SB-POREC-20110	Transaction_Type field in RCV_HEADERS_INTERFACE should be NEW or CANCEL.	Derive_Required_Columns/ Check_transaction_type
SB-POREC-20111	Validation_Flag field in RCV_HEADERS_INTERFACE should be Y or N.	Derive_Required_Columns/ Check_validation_flag
SB-POREC-20112	Auto_Transact_Code field in RCV_HEADERS_INTERFACE should be SHIP or RECEIVE or DELIVER.	Derive_Required_Columns/ Check_auto_transact_code
SB-POREC-20113	Receipt_Num cannot be NULL when Auto_Transact_Code is SHIP.	Derive_Required_Columns/ Check_receipt_num
SB-POREC-20114	Shipped_Date cannot be NULL when Asn_Type is ASN or ASBN.	Derive_Required_Columns/ Check_shipped_date
SB-POREC-20115	Invoice_Num cannot be NULL when Asn_Type is ASBN.	Derive_Required_Columns/ Check_invoice_num
SB-POREC-20116	Invoice_Date cannot be NULL when Asn_Type is ASBN.	Derive_Required_Columns/ Check_invoice_date
SB-POREC-20117	Total_Invoice_Amount cannot be NULL or negative when Asn_Type is ASBN.	Derive_Required_Columns/ Check_total_invoice_amount
SB-POREC-20118	Both Employee_Id and Employee_Name cannot be null when Auto_Transact_Code is RECEIVE.	Derive_Required_Columns/ Check_employee_name_id
SB-POREC-20119	Invalid Vendor_Name.	Derive_Required_Columns/ Check_Vendor_Name
SB-POREC20189	Shipped_Date cannot be earlier than SYSDATE or EXPECTED_RECEIPT_DATE.	Derive_Required_Columns/ Check_Shipped_Date_Range
SB-POREC20190	Invalid Employee_name.	Derive_Required_Columns/ Check_Validity_Employee_Name

Transactions

Table 44 sb_validate_rcv_trans_pkg

Error Code	Error Message	Generated from
SB-POREC-20120	Interface_Transaction_ID field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_Interface_Transaction_ID
SB-POREC-20121	Group_ID field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_group_id
SB-POREC-20122	Transaction_Type field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_transaction_type
SB-POREC-20123	Transaction_Date field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_transaction_date
SB-POREC-20124	Processing_Status_Code field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_processing_status_code
SB-POREC-20125	Processing_Mode_Code field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_processing_mode_code
SB-POREC-20126	Transaction_Status_Code field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_transaction_status_code
SB-POREC-20127	Quantity field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_quantity
SB-POREC-20128	Unit_Of_Measure field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_unit_of_mear
SB-POREC-20129	Item_Description field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_item_description
SB-POREC-20130	Value required in Document_Line_Num or Item_Num or Vendor_Item_Num or Item_Id or Po_Line_Id in rcv_transactions_interface table and currently all are null.	Validate_Required_Columns/ Check_document_line_num
SB-POREC-20131	Auto_Transact_Code field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_auto_transact_code
SB-POREC-20132	Receipt_Source_Code field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_receipt_source_code

Table 44 sb_validate_rcv_trans_pkg

Error Code	Error Message	Generated from
SB-POREC-20133	Vendor_Site_Id field is mandatory in rcv_transactions_interface table and current value is null.	Check_vendor_site_code
SB-POREC-20134	Value required in either Vendor_Name or Vendor_Num fields in rcv_transactions_interface table and currently both are null.	Validate_Required_Columns/ Check_vendor_name_num
SB-POREC-20135	Source_Document_Code field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_source_document_code
SB-POREC-20136	Value required in Document_Num or Po_Header_Id and currently both are null.	Validate_Required_Columns/ Check_document_num
SB-POREC-20137	Header_Interface_Id field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_header_interface_id
SB-POREC-20138	Validation_Flag field is mandatory in rcv_transactions_interface table and current value is null.	Validate_Required_Columns/ Check_validation_flag
SB-POREC-20139	FK Validation. Invalid Routing_Header_Id.	Validate_Foreign_Key_Checks/ Check_routing_header_id
SB-POREC-20140	FK Validation. Invalid Routing_Step_Id.	Validate_Foreign_Key_Checks/ Check_routing_step_id
SB-POREC-20141	FK Validation. Invalid Parent_Transaction_Id.	Validate_Foreign_Key_Checks/ Check_parent_transaction_id
SB-POREC-20142	FK Validation. Invalid Po_Header_Id.	Validate_Foreign_Key_Checks/ Check_po_header_id
SB-POREC-20143	FK Validation. Invalid Po_Line_Id.	Validate_Foreign_Key_Checks/ Check_po_line_id
SB-POREC-20144	FK Validation. Invalid Po_Line_Location_Id.	Validate_Foreign_Key_Checks/ Check_po_line_location_id
SB-POREC-20145	FK Validation. Invalid Inv_Transaction_Id.	Validate_Foreign_Key_Checks/ Check_inv_transaction_id
SB-POREC-20146	FK Validation. Invalid Po_Distribution_Id.	Validate_Foreign_Key_Checks/ Check_po_distribution_id
SB-POREC-20147	FK Validation. Invalid Po_Release_Id.	Validate_Foreign_Key_Checks/ Check_po_release_id
SB-POREC-20148	Foreign Key Validation. Invalid Requisition_Line_Id.	Validate_Foreign_Key_Checks/ Check_requisition_line_id
SB-POREC-20149	FK Validation. Invalid Requisition_Distribution_Id.	Validate_Foreign_Key_Checks/ Check_req_distribution_id
SB-POREC-20150	FK Validation. Invalid Inspection_Quality_Code.	Validate_Foreign_Key_Checks/ Check_inspection_quality_code

Table 44 sb_validate_rcv_trans_pkg

Error Code	Error Message	Generated from
SB-POREC-20151	FK Validation. Invalid Charge_Account_Id.	Validate_Foreign_Key_Checks/ Check_charge_account_id
SB-POREC-20152	FK Validation. Invalid Employee_Id.	Validate_Foreign_Key_Checks/ Check_employee_id
SB-POREC-20153	FK Validation. Invalid Wip_Entity_Id.	Validate_Foreign_Key_Checks/ Check_wip_entity_id
SB-POREC-20154	FK Validation. Invalid Item_Id.	Validate_Foreign_Key_Checks/ Check_item_id
SB-POREC-20155	FK Validation. Invalid Transportation_Account_Id.	Validate_Foreign_Key_Checks/ Check_tsrptn_account_id
SB-POREC-20156	FK Validation. Invalid Wip_Line_Id.	Validate_Foreign_Key_Checks/ Check_wip_line_id
SB-POREC-20157	FK Validation. Invalid Repetitive_Schedule_Id.	Validate_Foreign_Key_Checks/ Check_wip_rptv_schedule_id
SB-POREC-20158	Foreign Key Validation. Invalid Wip_Entity_Id for wip_operations.	Validate_Foreign_Key_Checks/ Check_wowip_entity_id
SB-POREC-20159	FK Validation. Invalid Wip_Entity_Id for wip_operations_resources.	Validate_Foreign_Key_Checks/ Check_worwip_entity_id
SB-POREC-20160	FK Validation. Invalid Bom_Resource_Id.	Validate_Foreign_Key_Checks/ Check_bom_resource_id
SB-POREC-20161	FK Validation. Invalid Department_Code.	Validate_Foreign_Key_Checks/ Check_department_code
SB-POREC-20162	FK Validation. Invalid Freight_Carrier_Code.	Validate_Foreign_Key_Checks/ Check_freight_carrier_code
SB-POREC-20163	FK Validation. Invalid SubInventory.	Validate_Foreign_Key_Checks/ Check_subinventory
SB-POREC-20164	FK Validation. Invalid Locator_Id.	Validate_Foreign_Key_Checks/ Check_locator_id
SB-POREC-20165	FK Validation. Invalid Uom_Code.	Validate_Foreign_Key_Checks/ Check_uom_code
SB-POREC-20166	FK Validation. Invalid Currency_Conversion_Type.	Validate_Foreign_Key_Checks/ Check_currency_conversion_type
SB-POREC-20167	FK Validation. Invalid Reason_Id.	Validate_Foreign_Key_Checks/ Check_reason_id
SB-POREC-20168	FK Validation. Invalid Source_Doc_Unit_Of_Measure.	Validate_Foreign_Key_Checks/ Check_source_doc_unit_of_measr
SB-POREC-20170	FK Validation. Invalid Shipment_Header_Id.	Validate_Foreign_Key_Checks/ Check_shipment_header_id
SB-POREC-20171	FK Validation. Invalid Shipment_Line_Id.	Validate_Foreign_Key_Checks/ Check_shipment_line_id
SB-POREC-20172	FK Validation. Invalid Primary_Unit_Of_Measure.	Validate_Foreign_Key_Checks/ Check_primary_unit_of_measure

Table 44 sb_validate_rcv_trans_pkg

Error Code	Error Message	Generated from
SB-POREC-20173	Value required in any one of Item_Category or Category_Id or Document_Line_Num or Po_Line_Id field in rcv_transactions_interface.	Derive_Required_Columns/ Check_item_category
SB-POREC-20174	Transaction_Type should be SHIP or RECEIVE.	Derive_Required_Columns/ Check_transaction_type
SB-POREC-20175	Processing_Status_Code should be PENDING.	Derive_Required_Columns/ Check_processing_status_code
SB-POREC-20176	Processing_Mode_Code should be BATCH.	Derive_Required_Columns/ Check_processing_mode_code
SB-POREC-20177	Auto_Transact_Code should be RECEIVE or DELIVER or SHIP.	Derive_Required_Columns/ Check_auto_transact_code
SB-POREC-20178	Receipt_Source_Code should be VENDOR.	Derive_Required_Columns/ Check_receipt_source_code
SB-POREC-20179	Source_Document_Code should be PO.	Derive_Required_Columns/ Check_source_document_code
SB-POREC-20180	Header_Interface_Id value not found in rcv_headers_interface.	Derive_Required_Columns/ Check_header_interface_id
SB-POREC-20181	Validation_Flag should be Y or N.	Derive_Required_Columns/ Check_validation_flag
SB-POREC-20182	Employee_Id should not be null when Transaction_Type is DELIVER.	Derive_Required_Columns/ Check_employee_id
SB-POREC-20183	Valid Group_Id not found in rcv_headers_interface.	Derive_Required_Columns/ Check_group_id
SB-POREC-20184	Invalid Ship_To_Location_Code or Ship_To_Location_Id when ASN_TYPE is ASN.	Derive_Required_Columns/ Check_ship_code_id
SB-POREC-20185	To_Organization_Code and To_Organization_Id, both cannot be null.	Derive_Required_Columns/ Check_org_code_id
SB-POREC-20186	There should be a valid Destination_Type_Code when Auto_Transact_Code is DELIVER.	Derive_Required_Columns/ Check_destination_type_code
SB-POREC-20187	Invalid Expected_Receipt_Date.	Derive_Required_Columns/ Check_expected_receipt_date
SB-POREC-20188	Transaction_Status_Code should be ERROR or COMPLETED.	Derive_Required_Columns/ Check_transaction_status_code
SB-POREC-20191	Invalid Item_Category.	Derive_Required_Columns/ Check_Valid_Item_Category
SB-POREC-20192	Invalid Location_Code.	Derive_Required_Columns/ Check_Valid_Loc_Code

Table 44 sb_validate_rcv_trans_pkg

Error Code	Error Message	Generated from
SB-POREC-20193	Invalid Vendor_Name.	Derive_Required_Columns/ Check_Valid_Vendor_Name
SB-POREC-20194	Invalid Vendor_Num.	Derive_Required_Columns/ Check_Valid_Vendor_Num

B.2.3 Requisitions

Table 45 sb_validate_por_req_pkg

Error Code	Error Message	Generated From
SB-POREQ-20100	Interface_Source_Code is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Interface_Source_Code
SB-POREQ-20101	Destination_Type_Code is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Destination_Type_Code
SB-POREQ-20102	Authorization_Status is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Authorization_Status
SB-POREQ-20103	Quantity is mandatory and the current value is NULL.	Validate_Required_Columns/ Check_Quantity
SB-POREQ-20104	Preparer_ID and Preparer_Name values are NULL.Either one should have value.	Validate_Required_Columns/ Check_Preparer_ID
SB-POREQ-20105	Charge_Account_ID and Charge_Account_Segment1 values are NULL.Either one should have value.	Validate_Required_Columns/ Check_Charge_Account_ID
SB-POREQ-20106	Destination_Organization_ID and Destination_Organization_Code values are NULL.Either one should have value.	Validate_Required_Columns/ Check_Dest_Org_ID
SB-POREQ-20107	Deliver_To_Location_ID and Deliver_To_Location_Code values are NULL.Either one should have value.	Validate_Required_Columns/ Check_Deli_To_Loc_ID
SB-POREQ-20108	Deliver_To_Requestor_ID and Deliver_To_Requestor_Name values are NULL.Either one should have value.	Validate_Required_Columns/ Check_Deli_To_Reqtor_ID
SB-POREQ-20109	If Currency_Code value present Rate_Date and Rate_Type are mandatory.	Validate_Required_Columns/ Check_Currency_Code
SB-POREQ-20110	If Project_Accounting_Context is Y then you must enter relevent project accounting info like project_id,task_id,expenditure_type and expenditure_org_id.	Validate_Required_Columns/ Check_Proj_Acct_Cont
SB-POREQ-20111	If Destination_Type_code is INVENTORY then you must enter Destination_Subinventory and Item_ID.	Validate_Required_Columns/ <ul style="list-style-type: none"> ▪ Check_Dest_Subin ▪ Check_Dest_Subinv
SB-POREQ-20112	If Source_Type_Code is INVENTORY then you must enter Item_ID.	Validate_Required_Columns/ Check_Src_Type_Code
SB-POREQ-20113	If Destination_Type_Code is EXPENSE or SHOP FLOOR then you must Source_Type_Code.	Validate_Required_Columns/ Check_Src_Type_Code1

Table 45 sb_validate_por_req_pkg

Error Code	Error Message	Generated From
SB-POREQ-20114	Requisition_Type value either INTERNAL or PURCHASE.	Validate_Required_Columns/ Check_Req_Type
SB-POREQ-20115	Requisition Header ID is invalid,FK validation failed against po_requisition_headers_all table.	Validate_Foreign_Key_Checks/ Check_Requisition_Header_ID
SB-POREQ-20116	Category ID is invalid,FK validation failed against mtl_categories_b table.	Validate_Foreign_Key_Checks/ Check_Category_ID
SB-POREQ-20117	Unit_Of_Measure is invalid,FK validation failed against mtl_units_of_measure_tl table.	Validate_Foreign_Key_Checks/ Check_Unit_Of_Measure
SB-POREQ-20118	Line_Type_ID is invalid,FK validation failed against po_line_types_b table.	Validate_Foreign_Key_Checks/ Check_Line_Type_ID
SB-POREQ-20119	UN_Number_ID is invalid,FK validation failed against po_un_numbers_b table.	Validate_Foreign_Key_Checks/ Check_UN_Number_ID
SB-POREQ-20120	Hazard_Class_ID is invalid,FK validation failed against po_hazard_classes_b table.	Validate_Foreign_Key_Checks/ Check_HAZARD_CLASS_ID
SB-POREQ-20121	WIP_Entity_ID is invalid,FK validation failed against wip_entities table.	Validate_Foreign_Key_Checks/ Check_WIP_Entity_ID
SB-POREQ-20122	WIP_Line_ID is invalid,FK validation failed against wip_lines table.	Validate_Foreign_Key_Checks/ Check_WIP_Line_ID
SB-POREQ-20123	WIP_Entity_ID or Dest_Org_ID or Wip_Operation_Seq_Num or Wip_Rept_Sch_ID either one is invalid,FK validation failed against wip_operations table.	Validate_Foreign_Key_Checks/ Check_WIP_Entity_ID_O
SB-POREQ-20124	WIP_Entity_ID or Dest_Org_ID or Wip_Operation_Seq_Num or Wip_Rept_Sch_ID or Resource_Seq_Num either one is invalid,FK validation failed against wip_operation_resources table.	Validate_Foreign_Key_Checks/ Check_WIP_Entity_ID_OR
SB-POREQ-20125	WIP_Repetitive_Schedule_ID or Dest_Org_ID either one is invalid,FK validation failed against wip_repetitive_schedules table.	Validate_Foreign_Key_Checks/ Check_WIP_Rept_Sch_ID
SB-POREQ-20126	Requisition_Line_ID is invalid,FK validation failed against po_requisition_lines_all table.	Validate_Foreign_Key_Checks/ Check_Requisition_Line_ID
SB-POREQ-20127	Src_Org_ID is invalid,FK validation failed against org_organization_definitions table.	Validate_Foreign_Key_Checks/ Check_Source_Organization_ID

Table 45 sb_validate_por_req_pkg

Error Code	Error Message	Generated From
SB-POREQ-20128	Dest_Org_ID is invalid,FK validation failed against org_organization_definitions table.	Validate_Foreign_Key_Checks/ Check_Dest_Org_ID_FK
SB-POREQ-20129	Source_Subinventory is invalid for a given Src_Org_ID,FK validation failed against MTL_Secondary_Inventories table.	Validate_Foreign_Key_Checks/ Check_Source_Subinventory
SB-POREQ-20130	Destination_Subinventory is invalid for a given Dest_Org_ID,FK validation failed against MTL_Secondary_Inventories table.	Validate_Foreign_Key_Checks/ Check_Dest_Subin_FK
SB-POREQ-20131	Deliver_To_Requestor_ID is invalid,FK validation failed against hr_employees table.	Validate_Foreign_Key_Checks/ Check_Del_To_Req_ID_FK
SB-POREQ-20132	Suggested_Buyer_ID is invalid,FK validation failed against po_agents table.	Validate_Foreign_Key_Checks/ Check_Suggested_Buyer_ID
SB-POREQ-20133	Suggested_Vendor_ID is invalid,FK validation failed against po_vendors table.	Validate_Foreign_Key_Checks/ Check_Suggested_Vendor_ID
SB-POREQ-20134	Suggested_Vendor_Site_ID is invalid,FK validation failed against po_vendor_sites_all table.	Validate_Foreign_Key_Checks/ Check_Suggested_Vendor_Site_ID
SB-POREQ-20135	Req_Distribution_ID is invalid,FK validation failed against Po_Req_Distributions_All table.	Validate_Foreign_Key_Checks/ Check_Req_Distribution_ID
SB-POREQ-20136	Suggested_Vendor_Contact_ID is invalid,FK validation failed against Po_Vendor_Contacts table.	Validate_Foreign_Key_Checks/ Check_Sug_Vendor_Con_ID
SB-POREQ-20137	Note1_ID is invalid,FK validation failed against Po_Notes table.	Validate_Foreign_Key_Checks/ Check_Note1_ID
SB-POREQ-20138	Accrual_Account_ID is invalid,FK validation failed against Gl_Code_Combinations table.	Validate_Foreign_Key_Checks/ Check_Accrual_Account_ID
SB-POREQ-20139	Variance_Account_ID is invalid,FK validation failed against Gl_Code_Combinations table.	Validate_Foreign_Key_Checks/ Check_Variance_Account_ID
SB-POREQ-20140	Budget_Account_ID is invalid,FK validation failed against Gl_Code_Combinations table.	Validate_Foreign_Key_Checks/ Check_Budget_Account_ID
SB-POREQ-20141	Ussgl_Transaction_Code is invalid,FK validation failed against gl_ussgl_transaction_codes table.	Validate_Foreign_Key_Checks/ Check_Ussgl_Transaction_Code

Table 45 sb_validate_por_req_pkg

Error Code	Error Message	Generated From
SB-POREQ-20142	Currency_Code is invalid,FK validation failed against fnd_currencies table.	Validate_Foreign_Key_Checks/ Check_Currency_Code
SB-POREQ-20143	Rate is invalid,FK validation failed against gl_daily_conversion_rates_r10 table.	Validate_Foreign_Key_Checks/ Check_Rate
SB-POREQ-20144	Rate_Type is invalid,FK validation failed against gl_daily_conversion_types table.	Validate_Foreign_Key_Checks/ Check_Rate_Type
SB-POREQ-20145	Approver_ID is invalid,FK validation failed against hr_employees table.	Validate_Foreign_Key_Checks/ Check_Approver_ID
SB-POREQ-20146	Autosource_Doc_Header_ID is invalid,FK validation failed against po_autosource_documents_all table.	Validate_Foreign_Key_Checks/ Check_Autosource_Doc_Header_ID
SB-POREQ-20147	Project_ID is invalid,FK validation failed against pa_projects_all table.	Validate_Foreign_Key_Checks/ Check_Project_ID
SB-POREQ-20148	Task_ID is invalid,FK validation failed against pa_tasks table.	Validate_Foreign_Key_Checks/ Check_Task_ID
SB-POREQ-20149	Note2_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note2_ID
SB-POREQ-20150	Note3_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note3_ID
SB-POREQ-20151	Note4_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note4_ID
SB-POREQ-20152	Note5_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note5_ID
SB-POREQ-20153	Note6_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note6_ID
SB-POREQ-20154	Note7_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note7_ID
SB-POREQ-20155	Note8_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note8_ID
SB-POREQ-20156	Note9_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note9_ID
SB-POREQ-20157	Note10_ID is invalid,FK validation failed against po_notes table.	Validate_Foreign_Key_Checks/ Check_Note10_ID
SB-POREQ-20158	Expenditure_Type is invalid,FK validation failed against pa_expenditure_types table.	Validate_Foreign_Key_Checks/ Check_Expenditure_Type
SB-POREQ-20159	Document_Type_Code is invalid,FK validation failed against po_document_types_all_b table.	Validate_Foreign_Key_Checks/ Check_Document_Type_Code

Table 45 sb_validate_por_req_pkg

Error Code	Error Message	Generated From
SB-POREQ-20160	Kanban_Card_ID is invalid,FK validation failed against mtl_kanban_cards table.	Validate_Foreign_Key_Checks/ Check_Kanban_Card_ID
SB-POREQ-20161	Preparer_ID is invalid,FK validation failed against hr_employees table.	Validate_Foreign_Key_Checks/ Check_Preparer_ID_FK
SB-POREQ-20162	Item_ID is invalid,FK validation failed against mtl_system_items_b table.	Validate_Foreign_Key_Checks/ Check_Item_ID
SB-POREQ-20163	Item_Revision is invalid,FK validation failed against mtl_item_revisions table.	Validate_Foreign_Key_Checks/ Check_Item_Revision
SB-POREQ-20164	Charge_Account_ID is invalid,FK validation failed against gl_code_combinations table.	Validate_Foreign_Key_Checks/ Check_Charge_Account_ID_FK
SB-POREQ-20165	If Dest_Type_Code is INVENTORY then item must be stock enabled for dest org. If Dest_Subinventory present then item must be valid either in subinventory or to a subinventory.	Perform_Other_Validations/ Check_Dest_Type_Code_Inv
SB-POREQ-20166	If Dest_Type_Code is EXPENSE then item must be purchasing enabled for purchasing and dest org.	Perform_Other_Validations/ Check_Dest_Type_Code_Exp
SB-POREQ-20167	If Dest_Type_Code is SHOP FLOOR then item must be an outise-operation item and purchasing enabled for purchasing and dest org.	Perform_Other_Validations/ Check_Dest_Type_Code_Shop
SB-POREQ-20168	If Src_Type_Code is INVENTORY then the item must be stock enabled for the src org and internal-order-enabled for the purchasing.	Perform_Other_Validations/ Check_Src_Type_Code_Inv
SB-POREQ-20169	If Src_Type_Code is VENDOR then the item must be purchasing enabled for the purchasing and dest org id.	Perform_Other_Validations/ Check_Src_Type_Code_Ven
SB-POREQ-20170	Deliver_To_Location_ID is invalid,FK validation failed against Hr_Locations table.	Validate_Foreign_Key_Checks/ Check_Deli_To_Loc_ID_FK
SB-POREQ-20171	category_id is invalid for the given item.	Perform_Other_Validations/ Check_Category_ID_Val

Index

A

addBatch method 127, 163
 Assigning ETDs to Event Types 46
 Autorun 22

C

CallableStatementAgent Class 144–160
 CallableStatementAgent constructor 145, 146
 cancel method 110
 Changing the User Name 72
 class parameter (Connector Settings) 106
 class parameter (Database Settings) 103
 clearBatch method 110
 clearParameters method 127
 clearWarnings method 111
 Collaboration 84, 86, 98
 Rules 84
 Service 84
 Collaboration Rules script 85
 Components, e*Way 18
 configuration
 Connector Settings 106–107
 DataSource Settings 103–105
 configuration parameters
 class (Connector Settings) 106
 class (Database Settings) 103
 connection establishment mode 107
 connection inactivity timeout 107
 connection verification interval 107
 DataBaseName 104
 Driver Type 103
 password 104
 PortNumber 104
 ServerName 104
 timeout 105
 transaction mode 106
 type 106
 user name 104
 configuration procedures 71
 connection establishment mode parameter 107
 connection inactivity timeout parameter 107
 Connection Manager 79
 connection verification interval 107

connector objects, JDBC 106
 Connector Settings 106–107
 Constructors 145–146, 162
 CallableStatementAgent 145, 146
 SqlStatementAgent 162
 conventions, writing in document 12
 COUNT method/procedure 88
 Creating a Schema 39

D

DatabaseName parameter 104
 DataSource Settings 103–105
 DELETE method/procedure 89
 driver class, JDBC 103
 Driver Type parameter 103
 driver type, JDBC 103

E

e*Way
 Components 18
 configuration 71
 creating 69
 Installation 22
 Properties 70
 Schedules 72
 Startup Options 72
 troubleshooting 84
 e*Way Connection configuration
 Connector Settings 106–107
 DataSource Settings 103–105
 e*Way Connections 76, 79
 Event Type 46
 Event Type Definition (ETD) 46, 86
 execute method 127, 163
 executeBatch method 111
 executeQuery method 128, 164
 executeUpdate method 128, 164

G

getArray method 158
 getBigDecimal method 154
 getBlob method 159
 getBoolean method 150
 getByte method 151
 getBytes method 158
 getClob method 159
 getDate method 154, 155
 getDouble method 153
 getFetchDirection method 112
 getFetchSize method 112

getFloat method 153
 getInt method 152
 getLong method 152
 getMaxFieldSize method 113
 getMaxRows method 113
 getMoreResults method 114
 getObject method 149, 150
 getQueryTimeout method 114
 getRef method 160
 getResultSet method 115
 getResultSetConcurrency method 115
 getResultSetType method 116
 getShort method 151
 getString method 157
 getTime method 155, 156
 getTimestamp method 156, 157
 getUpdateCount method 116
 getWarnings method 117

I

INITIALIZE method/procedure 90
 Installation
 e*Way 22
 InstallShield 22
 Intelligent Queue (IQ) 47, 84
 isClosed method 117

J

Java methods
 addBatch 127, 163
 cancel 110
 clearBatch 110
 clearParameters 127
 clearWarnings 111
 COUNT 88
 DELETE 89
 execute 127, 163
 executeBatch 111
 executeQuery 128, 164
 executeUpdate 128, 164
 getArray 158
 getBigDecimal 154
 getBlob 159
 getBoolean 150
 getByte 151
 getBytes 158
 getClob 159
 getDate 154, 155
 getDouble 153
 getFetchDirection 112
 getFetchSize 112
 getFloat 153

getInt 152
 getLong 152
 getMaxFieldSize 113
 getMaxRows 113
 getMoreResults 114
 getObject 149, 150
 getQueryTimeout 114
 getRef 160
 getResultSet 115
 getResultSetConcurrency 115
 getResultSetType 116
 getShort 151
 getString 157
 getTime 155, 156
 getTimestamp 156, 157
 getUpdateCount 116
 getWarnings 117
 INITIALIZE 90
 isClosed 117
 MOVE 91
 queryDescription 118
 queryName 118
 registerOutParameter 147, 148
 REQUEST 92
 REQUEST_STATUS 92
 resetRequested 119
 resultSetConcurToString 119
 resultSetDirToString 120
 resultSetTypeToString 120
 sessionClosed 121
 sessionOpen 121, 129, 147
 setArray 142
 setAsciiStream 140
 setBigDecimal 135
 setBinaryStream 140
 setBlob 142
 setBoolean 132
 setByte 132
 setBytes 139
 setCharacterStream 141
 setClob 143
 setCursorName 122
 setDate 136
 setDouble 135
 setEscapeProcessing 122
 setFetchDirection 123
 setFloat 134
 setInt 133
 setLong 134
 setMaxFieldSize 124
 setMaxRows 123
 setNull 129, 130
 setObject 130, 131
 setQueryTimeout 124

setRef 143
 setShort 133
 setString 139
 setTime 137
 setTimestamp 138
 stmtInvoke 125
 VALIDATE 93
 wasNull 149
 Java Object Classes
 CallableStatementAgent 144–160
 PreparedStatementAgent 126–143
 SqlStatementAgent 161–164
 StatementAgent 109–125
 JDBC
 connector objects 106
 driver class 103
 driver type 103

L

logging options 74

M

Mixing XA-Compliant and XA-Noncompliant e*Way Connections 106
 monitoring thresholds 75
 MOVE method/procedure 91

O

Object Classes 108–??
 ODBC Test 31

P

Participating Host 84
 password parameter 104
 PortNumber parameter 104
 PreparedStatementAgent Class 126–143
 Properties, e*Way 70
 publish 84

Q

queryDescription method 118
 queryName method 118
 Queues 47

R

Registering the DataSource 30
 registerOutParameter method 147, 148

REQUEST method/procedure 92
 REQUEST_STATUS method 92
 resetRequested method 119
 resultSetConcurToString method 119
 resultSetDirToString method 120
 resultSetTypeToString method 120

S

sb_validate_cc_entries_pkg 170, 217
 sb_validate_ci_xrefs_pkg 168, 215
 sb_validate_cust_items_pkg 165, 213
 sb_validate_invrev_pkg 178, 223
 sb_validate_items_pkg 172, 219
 sb_validate_poi_hdrs_pkg 187, 230
 sb_validate_poi_lines_pkg 191, 233
 sb_validate_poi_pkg 187, 230, 237
 sb_validate_por_req_pkg 206, 244
 sb_validate_rcv_hdr_pkg 196, 237
 sb_validate_rcv_pkg 196, 237
 sb_validate_rcv_trans_pkg 199, 239
 sb_validate_serial_nos_pkg 186, 229
 sb_validate_trans_lots_pkg 186, 229
 sb_validate_trans_pkg 183, 227
 Schedules 72
 Schema, creating 39
 ServerName parameter 104
 sessionClosed method 121
 sessionOpen method 121, 129, 147
 setArray method 142
 setAsciiStream method 140
 setBigDecimal method 135
 setBinaryStream method 140
 setBlob method 142
 setBoolean method 132
 setByte method 132
 setBytes method 139
 setCharacterStream method 141
 setClob method 143
 setCursorName method 122
 setDate method 136
 setDouble method 135
 setEscapeProcessing method 122
 setFetchDirection method 123
 setFloat method 134
 setInt method 133
 setLong method 134
 setMaxFieldSize method 124
 setMaxRows method 123
 setNull method 129, 130
 setObject method 130, 131
 setQueryTimeout method 124
 setRef method 143
 setShort method 133

Index

setString method 139
setTime method 137
setTimestamp method 138
Setting Startup Options or Schedules 72
SqlStatementAgent Class 161–164
SqlStatementAgent constructor 162
Staging Table Node 88
Startup Options 72
StatementAgent Class 109–125
stmtInvoke method 125
stored procedures
 COUNT 88
 DELETE 89
 INITIALIZE 90
 MOVE 91
 REQUEST 92
 VALIDATE 93
subscribe 84

T

timeout parameter 105
transaction mode parameter 106
troubleshooting the e*Way 84
type parameter 106

U

User name 72
user name parameter 104

V

VALIDATE method/procedure 93

W

wasNull method 149

X

XA 103