ORACLE NETSUITE

Electronic Invoicing

2022.2

March 22, 2023

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document may change and remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility.

Access to Oracle Support

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

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is" and “as available”, for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer’s use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle’s other customers, Oracle may slow down or throttle Customer’s excess use until such time that Customer’s use stays within reasonable limits. If Customer’s particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer’s actual usage pattern.

Beta Features

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

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

Send Us Your Feedback

We'd like to hear your feedback on this document.

Answering the following questions will help us improve our help content:

- Did you find the information you needed? If not, what was missing?
- Did you find any errors?
- Is the information clear?
- Are the examples correct?
- Do you need more examples?
- What did you like most about this document?

Click here to send us your comments. If possible, please provide a page number or section title to identify the content you're describing.

To report software issues, contact NetSuite Customer Support.

Table of Contents

Electronic Invoicing

Note: The Electronic Invoicing SuiteApp provides the ability to generate and send e-documents as well as receive and convert them into transactions records. Inbound and outbound electronic invoicing processes support multiple transaction records. For more information about supported transactions, see Transactions and Processes Supported by the Electronic Invoicing SuiteApp.

To understand how to set up and use the electronic invoicing features, read the following topics:

Electronic Invoicing Overview

The Electronic Invoicing SuiteApp enables you to create and use electronic documents (e-documents) in XML or JSON, for your business. The use of XML or JSON e-documents has become a standard for exchanging business information in many countries. E-documents can help you comply with legal requirements in your country, or to automate your order-to-cash and purchase-to-payment processes. By using e-documents, your company, customers, vendors, tax agencies or government regulatory bodies, can exchange information about business transactions electronically using standard data format.

The Electronic Invoicing SuiteApp supports outbound and inbound processing of e-documents. Outbound e-document processing involves generating XML or JSON e-documents from supported NetSuite transactions, and then sending the XML or JSON e-documents to your customers, vendors and tax agencies. On the other hand, inbound e-document processing involves receiving XML e-documents from your vendors and then converting the XML e-documents into NetSuite transaction records. Both inbound and outbound e-documents use templates that you can create. Sample templates for inbound and outbound e-documents are included in the SuiteApp.

Important: The Electronic Invoicing SuiteApp provides a framework for automating e-documents processing. It does not include native support for any country-specific requirements or e-document standards. But you can create custom country-specific e-document templates and packages using the Electronic Invoicing SuiteApp.

The Electronic Invoicing SuiteApp enables you to track received, converted, generated and sent e-documents through an audit trail on the **E-Document** subtab of a transaction record.

The SuiteApp is available free of charge when used for a single country. To use e-documents across multiple countries, contact your NetSuite account representative to purchase a license. For more information, see Electronic Invoicing SuiteApp Availability and License Client.

The Electronic Invoicing SuiteApp also supports multiple languages that the NetSuite user interface can be displayed in. For more information, see the help topics Configuring Multiple Languages and Choosing a Language for Your NetSuite User Interface.

Outbound Electronic Invoicing

The Electronic Invoicing SuiteApp enables you to generate outbound XML or JSON e-documents from the following NetSuite transactions:

- Bill
- Credit Memos
- Cash Sales
- Cash Refunds
- Customer Payments
- Estimates

- Invoices
- Item Fulfillment
- Purchase Orders
- Registered Custom Transaction Types
- Returns
- Transfer Order
- Vendor Credit or Bill Credit

Generated e-documents can then be sent to your customers, vendors or tax agency, individually or in batches. The default sending method of the outbound electronic invoicing is through email. You can also create custom sending methods like SOAP web services or automate the sending of e-documents by deploying scripts. The electronic invoicing outbound process also supports e-document certification, by sending e-documents to a certification authority and then receiving the certified e-documents.

In general, the electronic invoicing outbound process is composed of the following steps:

1. The administrator creates and sets up outbound e-document templates and sending methods:
    a. The outbound e-document template must contain the XML or JSON code and specify the supported transactions and the recipient subsidiaries. For more information, see Creating E-Document Templates and Multi-subsidiary Support in the Outbound Process.
    b. The sending method must specify the custom plug-in implementation, the supported transactions, the recipient subsidiaries, and if it will be used for certification. For more information, see Creating E-Document Sending Methods, E-Document Certification in the Outbound Process, and Multi-subsidiary Support in the Outbound Process.
2. The administrator creates an e-document package, which specifies an e-document template and sending method. See Creating E-Document Packages. Users can assign the e-document package to customers or vendors. The transactions of that customer or vendor will apply the e-document template and sending method of the e-document package. For more information, see Assigning E-Document Packages to Customer or Vendor Records.

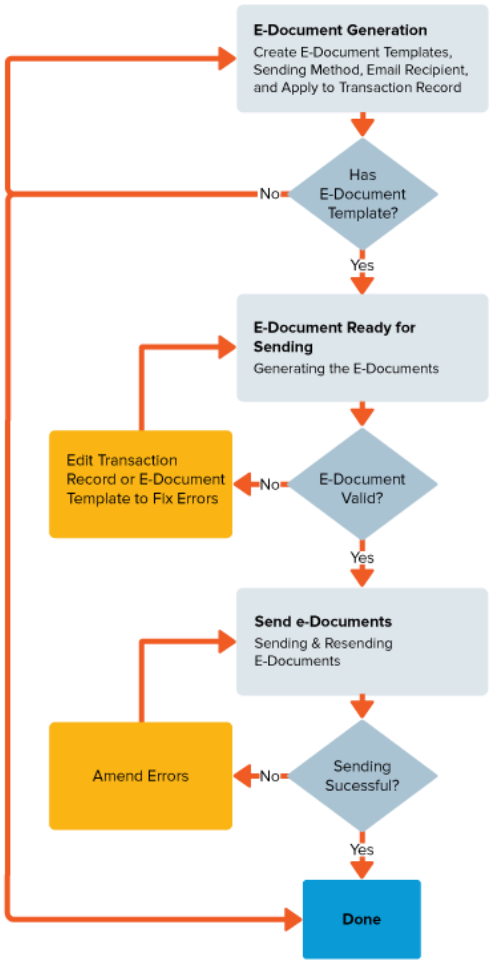
    Or, users can create or edit a transaction record, then select an e-document template and an e-document sending method. For more information, see Transactions and Processes Supported by the Electronic Invoicing SuiteApp and Selecting E-Document Packages, Templates and Sending Methods on Transactions.
3. Users can generate the outbound e-documents of transactions with an associated e-document template and sending method. Outbound e-documents can be generated from the following transaction types:
    - Bill
    - Credit Memos
    - Cash Sales
    - Cash Refunds
    - Customer Payments
    - Estimates
    - Invoices
    - Item Fulfillment
    - Purchase Orders
    - Returns
    - Registered Custom Transaction Types

- Transfer Order
- Vendor Credit or Bill Credit

See Generating and Regenerating E-Documents.

4. Users can send generated e-documents to your customers, vendors or tax agency. For more information, see Sending and Resending E-Documents.
    a. If you need to certify the generated e-document, specify a certification sending method. Then, send the e-documents to the certification authority or tax agency. See E-Document Certification in the Outbound Process.
5. Users can perform all the e-document processes by clicking the single **Process E-Document** button in a transaction , if the e-document template and sending method is configured. See Processing E-Documents Automatically for Individual Transactions

The following diagram shows the general process flow of outbound electronic invoicing in NetSuite.



For more information about the process of generating and sending outbound e-documents, read the following topics:

- Overview of Outbound E-Document Process
- Selecting E-Document Packages, Templates and Sending Methods on Transactions
- Generating and Regenerating E-Documents

- Sending and Resending E-Documents

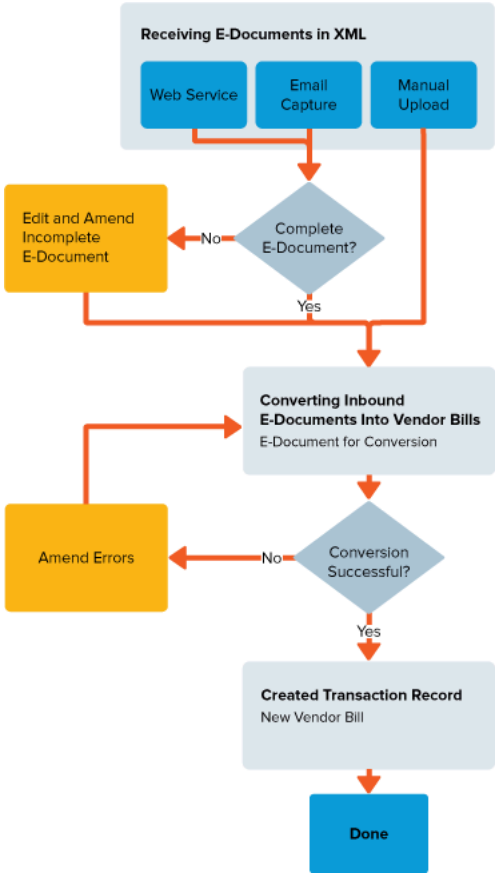
Inbound Electronic Invoicing

You can receive inbound e-documents from your vendors or government regulatory bodies. Inbound e-documents are received in NetSuite through email or SOAP web services, which are the supported inbound channels. You can also receive e-documents from portable file storage media or devices from your vendors. Then, you can manually upload to NetSuite the received XML e-documents, which you will subsequently convert into NetSuite transactions.

An inbound e-document template must be created to convert the received e-documents into supported NetSuite transactions. A sample inbound e-document template is included with the SuiteApp.

For example, a vendor whom you initially sent a purchase order may send you an XML e-document invoice, which you receive and then convert into a vendor bill record in NetSuite.

The following diagram shows the general process flow of inbound electronic invoicing in NetSuite.



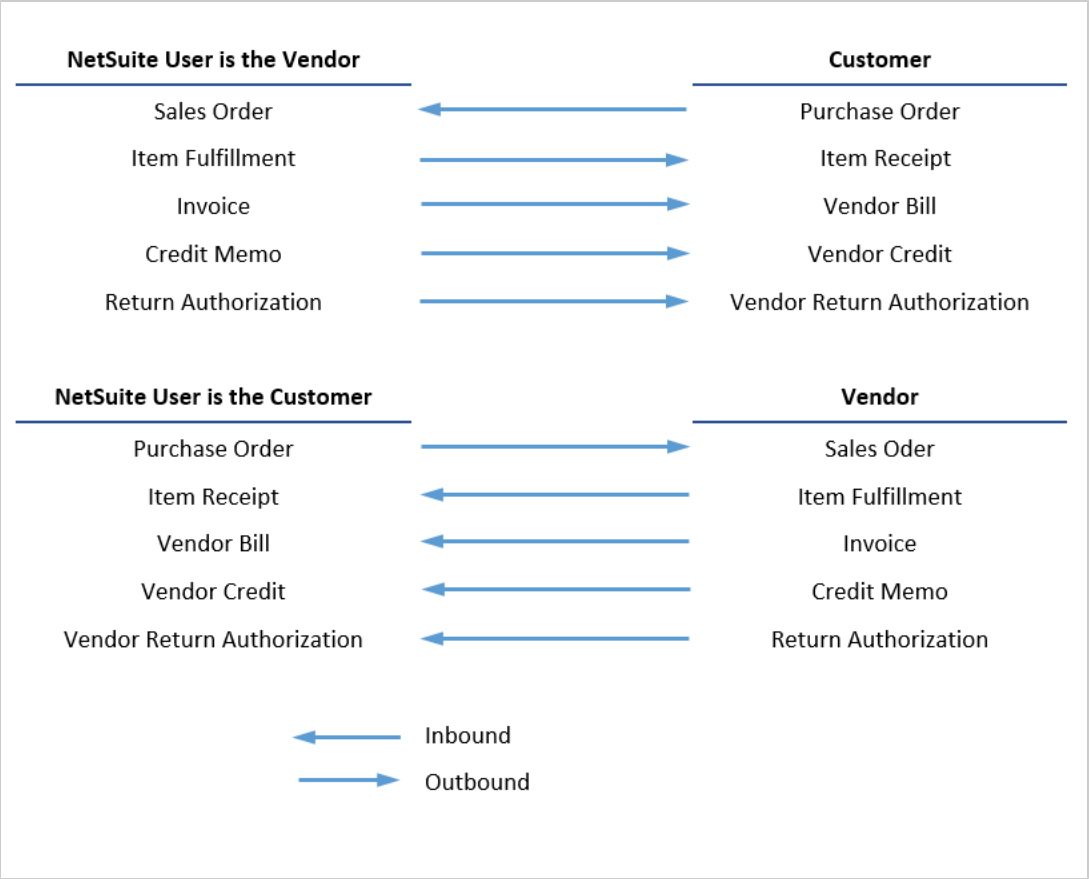
For more information about the process of receiving and converting inbound e-documents, read the following topics:

- Overview of Inbound E-Document Processing
- Receiving Inbound E-Documents by Email Capture
- Uploading Received XML Files as Inbound E-Documents
- Converting Inbound E-Documents into Transaction Records

Transactions and Processes Supported by the Electronic Invoicing SuiteApp

Inbound and outbound transaction types are better understood in the perspective of the NetSuite user, who can either be the vendor or the customer in the conduct of a transaction.

The following diagram illustrates scenarios where transaction types are transformed into other types in the perspective of the NetSuite user along the outbound and inbound e-document process.



Based on the diagram, the NetSuite user can either be a vendor or a customer, who can generate and send XML or JSON outbound e-documents or receive inbound XML e-documents using the Electronic Invoicing SuiteApp.

As the vendor (the party selling items or providing services), the NetSuite user can generate the XML or JSON e-document of an invoice, based on an initial purchase order sent by a customer. Likewise, the NetSuite user can generate the XML or JSON e-document of an item fulfillment, credit memo, or return authorization. Generated outbound XML or JSON e-documents can then be sent through email or web service to a customer or tax authority.

Outbound e-document generation for vendor bill is a special case, as it supports self-billing. This is useful if you want to charge your company in advance for goods or services acquired from a vendor, and then send the generated e-document of the bill to your vendor along with the payment.

As the customer (the party buying items or acquiring services), the NetSuite user can receive inbound XML e-documents from a vendor or tax authority through email or web service. After the inbound XML e-document is received in NetSuite, it can be converted into its corresponding transaction record. A vendor

can send to your company the e-document of an invoice, which you can receive and convert into a vendor bill. Currently, only vendor bill is supported for inbound processing.

The following table lists the transaction types currently supported by outbound and inbound electronic invoicing.

Outbound	Inbound
Bill (for self-billing)	Bill
Credit Memo	
Cash Refund	
Cash Sale	
Customer Payment	
Invoice	
Item fulfillment	
Purchase Order	
Return Authorization	
Registered Custom Transaction Types	
Transfer Order	
Vendor Credit or Bill Credit	

Electronic Documents Dashboard SuiteApp Portlet

The Electronic Documents Dashboard SuiteApp Portlet displays the number of e-documents that are pending a process. Clicking the number under a process opens a results page with a list of e-documents pending a process, or the page for performing an e-document process.

For outbound e-document processing, the left column displays the number of:

- Outbound E-Documents for Generation
- Outbound E-Documents for Sending, or Outbound E-Documents for Certification

  Note: If a certification sending method is specified, Outbound E-Documents for Certification is displayed instead of Outbound E-Documents for Sending.
- Certified E-documents for Sending is only displayed if at least one certification sending method is specified.
- Outbound E-Documents with Errors

  A Send Failed Outbound E-Documents link is included on the outbound column to allow searching and resending of outbound e-documents that failed initial sending. For more information, see Sending and Resending E-Documents.

For inbound e-document processing, the right column displays the number of:

- Inbound E-Documents for Conversion
- Convert Failed Inbound E-Documents

- Incomplete Inbound E-Documents

  The inbound column also has a link for manually uploading inbound e-documents. For more information, see Uploading Received XML Files as Inbound E-Documents.

E-document settings for administrator tasks can be accessed from the menu, Setup > E-Documents.

For more information about the e-documents portlet, see Granting Access Permission to the E-Documents Portlet and Displaying the E-Documents Portlet on the Home Page.

Customizations Supported by the Electronic Invoicing SuiteApp

The Electronic Invoicing SuiteApp is designed to support the following customizations:

- Custom sending methods – See Creating E-Document Sending Methods.
- Scheduling of outbound e-document generation – See Deploying the Bulk Generation Script for E-Documents.
- Scheduling of outbound e-document sending - See Deploying the Script for Scheduled Sending of E-Documents
- Scheduling of inbound e-document conversion - See Deploying Automatic Bulk Conversion Script for Inbound E-Documents
- Customizing the preferences for processing e-documents automatically – See Processing E-Documents Automatically for Individual Transactions

E-Document Audit Trail and Statuses

The **E-Document Audit Trail** subtab logs all changes to an e-document. The Audit Trail is available on the **E-Document** subtab of a transaction record. The e-document status describes the current processing state of a transaction or e-document. E-document statuses are indicated in the **E-Document Status** field on the **E-Document** subtab. For more information about e-document statuses, see Outbound E-Document Statuses and Inbound E-Document Statuses.

To understand e-document processing, including inbound and outbound process flows, see the following topics:

- Overview of Outbound E-Document Process
- Overview of Inbound E-Document Processing

Electronic Invoicing Permissions and Access Levels

The following tables show the permissions and access levels for features provided by the Electronic Invoicing SuiteApp:

Administrator and Setup Tasks

Tasks or Functionality	Role and Permission
Creating E-Document Packages	By default, only the Administrator can create, edit, or delete e-document packages. All other roles can only view e-document packages.

Tasks or Functionality	Role and Permission
Creating E-Document Templates	By default, only the Administrator can create, edit, or delete e-document template records. All other roles can only view e-document template records.
Creating E-Document Sending Methods Creating a Custom Plug-in Implementation for Sending E-Documents	By default, only the Administrator can create, edit, or delete e-document sending methods. All other roles can only view e-document sending methods.
Selecting a Designated E-Document Sender	By default, only the Administrator can select a designated e-document sender.
Assigning E-Document Packages to Customer or Vendor Records	Roles with access to customer and vendor and contact records can assign e-document packages to customer and vendor records.
Creating a Custom Plug-in for Inbound E-Document Validation	By default, only the Administrator can create a custom plug-in for inbound e-document validation.
Processing E-Documents Automatically for Individual Transactions	By default, only the Administrator can select the required **E-Document Automation Type** from the Automatic E-Invoicing tab on the E-Document Preferences page for an individual transaction.

Outbound Processing Permissions and Access Levels

Tasks or Functionality	Role and Permission
Defining E-Document Email Recipients	Roles with **Edit** or **Full** access to customer and vendor contact records can define e-document email recipients on customer or vendor records.
Setting Up Custom Roles to Send E-Documents Customizing Roles to Restrict E-Document Generation or Sending	The Administrator can restrict the default permission of roles to generate or send outbound e-documents.
Generating E-Documents for Single Transactions Regenerating E-Documents for Single Transactions	Roles with minimum permission to view supported transaction records, can generate and regenerate e-documents for single transactions. An administrator can grant custom roles the permission to generate e-document for sending to customers and vendors.
Deploying the Bulk Generation Script for E-Documents	Only the Administrator role can configure the script used to generate and regenerate e-documents in bulk.
Sending the E-Document of a Single Transaction Resending the E-Document of a Single Transaction	Roles with minimum permission to view supported transaction records, can send and resend e-documents for single transactions.
Setting Up Custom Roles to Send E-Documents Resending E-Documents in Bulk	The following roles can send and resend e-documents in bulk: - A/P Clerk - A/R Clerk - Accountant - Administrator - Bookkeeper - CFO

Tasks or Functionality	Role and Permission
	An administrator can grant custom roles the permission to send bulk e-documents. An administrator can grant custom roles the permission to send e-documents to customers/vendors.
Template and Sending Method Auto-selection For more information about Template and Sending Method Auto-selection, see Multi-subsidiary Support in the Outbound Process	Roles with **Edit** or **Full** access to customer and vendor records can have access to template and sending method auto-selection feature.

Inbound Processing Permissions and Access Levels

Tasks or Functionality	Role and Permission
Setting Up Custom Roles that can Convert Inbound E-Documents	Only the Administrator role can set the permission to perform conversion of inbound e-documents
Uploading Received XML Files as Inbound E-Documents	The following roles can upload inbound e-documents manually: ■ A/P Clerk ■ A/R Clerk ■ Accountant ■ Bookkeeper ■ CFO ■ Administrator ■ Custom Roles (with permission)
Converting Individual Inbound E-Documents into Vendor Bills Converting an Inbound E-Document Without a Purchase Order Number	The following roles can convert inbound e-documents into transaction records: ■ A/P Clerk ■ A/R Clerk ■ Accountant ■ CFO ■ Administrator ■ Custom Roles (with permission)
Converting Failed Inbound E-Documents Deploying Automatic Bulk Conversion Script for Inbound E-Documents	The following roles can convert inbound e-documents into transaction records in bulk: ■ A/P Clerk ■ A/R Clerk ■ Accountant ■ Bookkeeper ■ Buyer ■ CFO ■ Administrator ■ Custom Roles (with permission)
Canceling Inbound E-Documents	The following roles can cancel inbound e-documents: ■ A/P Clerk ■ A/R Clerk

Tasks or Functionality	Role and Permission
	■ Accountant ■ Bookkeeper ■ Buyer ■ CFO ■ Administrator ■ Custom Roles (with permission)
Step 4 of Prerequisites for Using Electronic Invoicing	Only the Administrator role can designate an employee or group of employees who will receive the email notification upon completion of batch conversion.

Vendor Credit or Bill Credit Permissions and Access Levels

Descriptions	Required Minimum Permissions	Access Levels
For Custom roles that can only create Vendor Credit or Bill Credit transactions but do not perform any electronic invoicing tasks.	E-document sending method > Custom Record subtab	View
	Perform search > List subtab	View
For Custom roles that can view or edit Vendor Credit or Bill Credit transactions but do not perform any electronic invoicing tasks.	E-document sending method > Custom Record subtab	View
	E-document templates > Custom Record subtab	View
	Perform search > List subtab	view
For custom Roles that can create, edit or view Vendor Credit or Bill Credit transactions and also perform the following electronic invoicing tasks: ■ Allow Manual generation of E-documents ■ Allow sending of E-documents for certification ■ Allow sending of E-documents	E-document sending method > Custom Records subtab	View
	E-document templates > Custom Records subtab	View
	E-document audit trail > Custom Records subtab	Full/Edit
	Setup Company > Setup subtab	View
	Vendor record > List subtab	View
	Perform Search > List subtab	View
	Find transaction > Transaction subtab	View

Electronic Invoicing SuiteApp Availability and License Client

The Electronic Invoicing SuiteApp enables you to use and manage e-documents for various transaction types. The SuiteApp has two major components, inbound and outbound e-document processing. Outbound e-document processing enables you to generate and send e-documents to other parties such as customers, vendors, or tax agencies. Inbound e-document processing enables you to receive e-documents from vendors and then convert them into bills. For more information, see Transactions and Processes Supported by the Electronic Invoicing SuiteApp.

The Electronic Invoicing SuiteApp can be installed for free in any NetSuite account.

For free use without a paid license in OneWorld accounts, you must select a country in the **E-Document Country for Free Use** field on the Company Information page from Setup > Setup Tasks > Company > Company Information. You must ensure that the country you select is:

- The same country where you will send e-documents to
- The same country of the vendor's subsidiary

However, if you want to send e-documents to multiple countries and convert e-documents received from other countries, you must purchase a license from your NetSuite account manager.

The NetSuite SuiteApps License Client (Bundle ID: 116144) must be installed before you install the Electronic Invoicing SuiteApp (Bundle ID: 436209).

The NetSuite SuiteApps License Client controls an account's access to the electronic invoicing features. Upon installation, the NetSuite SuiteApps License Client communicates with the NetSuite SuiteApps License Server to obtain active license information. If you have an active license, the E-Document Country for Free Use can be left blank.

You cannot use key features in inbound and outbound e-document processing such as sending e-documents to multiple countries and converting received e-documents from other countries if:

- The NetSuite SuiteApps License Client is not installed, or
- The NetSuite SuiteApps License Client is installed but your license is expired in OneWorld accounts.

Note: It takes a maximum of 30 minutes to update license status of Electronic Invoicing SuiteApp after license refresh.

For more information, see the help topic NetSuite SuiteApps License Client.

Electronic Invoicing Limitations and Best Practices

Read the following topics to know the current limitations of the Electronic Invoicing SuiteApp, and to be guided by best practices for setting up and using electronic invoicing features.

Electronic Invoicing Limitations

General limitations of the Electronic Invoicing SuiteApp:

- Only FreeMarker is supported for outbound e-document templates.
- Only the following transaction types are currently supported:
  - Bill
  - Cash sales
  - Cash refunds
  - Credit memos
  - Customer payment
  - Estimates
  - Invoices
  - Item Fulfillment
  - Purchase orders
  - Returns
  - Registered Custom Transaction Types

  - Transfer Order
  - Vendor Credit or Bill Credit
- Mass download of e-documents is not supported.
- Intercompany transfer order is not supported.
- Creating the following records using the create new icon is not supported.
  - E-Document Template (on all transaction records)
  - E-Document Package (on vendor/customer records, transfer order transaction records and basic and journal style custom transaction types)
  - E-Document Sending Method (on all transaction records)
  - E-Document Email Recipient (on transfer order transactions and basic and journal style custom transaction types)

Limitations in outbound e-document processing:

The system can send an e-document by email to a maximum of 10 recipients for each customer.

Note: The system counts each contact added as a recipient. If you add the same contact multiple times, each instance is considered one recipient.

Limitations in inbound e-document processing:

- Only one XML file and one PDF file reference can be processed per email received.
- Inbound e-documents can only be uploaded manually, one at a time.
- Currently, only vendor bills can be created or converted from inbound e-documents.

Electronic Invoicing Best Practices

Perform the following best practices to prevent errors and performance issues:

- Do not create or add MR script deployments. Maintain the default single script deployments of the Electronic Invoicing SuiteApp, which are: Generate E-Document Content MR, Automatic Send E-Document MR, and Convert Inbound E-Document MR.
- When assigning an e-document package to a customer or vendor, make sure that the customer or vendor has contact records if the e-document package uses an email sending method. To prevent validation errors caused by missing e-document email recipients, it is that you create the contact records first before assigning the e-document package to the customer or vendor.
- When using CSV import to assign e-document packages to customers or vendors, be sure to include email recipients for e-document packages that use an email sending method. The system will encounter an error when sending an e-document that has no email recipient.
- When sending e-documents in bulk, NetSuite that you first make sure that custom roles have the required permissions and access to use the bulk sending feature. See Setting Up Custom Roles to Send E-Documents.
- When sending e-documents in bulk, NetSuite that you use the filters to limit the number of e-documents to process at a time.
- When setting up sending method custom plug-ins:
  - There is no need to include loading of customer and invoice records as well as vendor and purchase order records to retrieve data within your custom plug-in. This information is already provided by the Electronic Invoicing SuiteApp.
  - There is no need to load recipients within your custom plug-in . This information is already provided by the Electronic Invoicing SuiteApp.

- Avoid loading e-document recipients within your custom plug-in for sending email. When an e-document package with an email sending channel is assigned to a customer or vendor, the Electronic Invoicing SuiteApp automatically performs a validation check to make sure customer or vendor records have recipients with valid email addresses. If your custom plug-in loads e-document recipients, those recipients will not be included in the validation.
- Be aware of SuiteScript 2.0 governance and time limits. Suitelet limits apply to individual sending. Map function limits (Map/Reduce script type) apply to bulk sending. See the following topics:
  - SuiteScript 2.x Suitelet Script Type
  - Map/Reduce Governance
- Test your e-document sending plug-ins thoroughly before using them on live data.
- Avoid data leaks when sending e-documents by email or sending to third party SOAP web services. Data leaks can be avoided by making sure codes are properly reviewed. Also make sure you are not sending more information than what is required.
- Use appropriate and specific error messages in your scripts. Error messages should inform users when a problem occurs, help users understand why the problem occurred, and an action that can fix the problem.

Electronic Invoicing Administrator Guide

Only the Administrator role can install the Electronic Invoicing SuiteApp and set up the custom records and templates required for generating and sending e-documents. The administrator must also deploy the script for generating e-documents in bulk.

The following topics are intended for administrators:

General Setup Tasks

- Installing the Electronic Invoicing SuiteApp
- Prerequisites for Using Electronic Invoicing
- Considerations When Setting Up Electronic Invoicing
- Electronic Invoicing Setup Tasks
- Defining the E-Document Country for Free Use
- Advanced PDF/HTML Template
- Granting Access Permission to the E-Documents Portlet
- Understanding E-Documents and E-Document Packages
- Creating E-Document Packages
- Multi-subsidiary Support in the Outbound Process
- Creating E-Document Templates
- Editing E-Document Templates

Outbound E-Document Processing Setup Tasks

- E-Document Certification in the Outbound Process
- Creating E-Document Sending Methods
- Setting Up an Email Sending Method for E-Documents
- Creating Custom Methods for Sending E-Documents
- Selecting a Designated E-Document Sender
- Setting Up Custom Roles to Send E-Documents

- Customizing Roles to Restrict E-Document Generation or Sending
- Deploying the Bulk Generation Script for E-Documents
- Deploying the Script for Scheduled Sending of E-Documents
- Updating E-Document Certification Statuses
- Processing E-Documents Automatically for Individual Transactions

Inbound E-Document Processing Setup Tasks

- Electronic Invoicing Inbound Email Capture
- Using SOAP Web Services for Inbound Processing
- Setting Up Custom Roles that can Convert Inbound E-Documents
- Inbound Validation Plug-ins
- Deploying Automatic Bulk Conversion Script for Inbound E-Documents

See also the following topics:

- Electronic Invoicing Overview
  - Understanding E-Documents and E-Document Packages
  - Electronic Invoicing Permissions and Access Levels
  - Electronic Invoicing Limitations and Best Practices
- Electronic Invoicing Errors
  - Electronic Invoicing Error Codes
  - Outbound E-Document Generation Errors
  - Outbound E-Document Sending Errors

Installing and Setting Up Electronic Invoicing

Read the following topics to understand how to install and set up Electronic Invoicing:

- Prerequisites for Using Electronic Invoicing
- Installing the Electronic Invoicing SuiteApp
- Considerations When Setting Up Electronic Invoicing
- Electronic Invoicing Setup Tasks
- Electronic Invoicing Limitations and Best Practices

Prerequisites for Using Electronic Invoicing

Before installing the Electronic Invoicing SuiteApp, you must complete the following steps:

1. Install the NetSuite SuiteApps License Client (Bundle ID: 116144). For more information, see the help topic NetSuite SuiteApps License Client.

   The NetSuite SuiteApps License Client (Bundle ID: 116144) must be installed before you install the Electronic Invoicing SuiteApp (Bundle ID: 436209). To install, see the help topic Installing a Bundle.

   The NetSuite SuiteApps License Client controls an account's access to the Electronic Invoicing features. Upon installation, the NetSuite SuiteApps License Client communicates with the NetSuite SuiteApps License Server to obtain active license information.

   If the account has an active license, you can use Electronic Invoicing features for multiple countries. If the account does not have an active license, you can still use the Electronic Invoicing

SuiteApp for free, but your account can generate and send e-documents to only one country, and convert into vendor bills received e-documents from the same country. See Defining the E-Document Country for Free Use to set up your account for free use of the Electronic Invoicing SuiteApp.

If the NetSuite SuiteApps License Client is not installed, your OneWorld accounts can neither generate and send e-documents nor convert received e-documents into vendor bills.

If the NetSuite SuiteApps License Client is installed, your single instance accounts can generate, certify or send all outbound e-documents and convert inbound e-documents into bills.

2. Make sure the following features are enabled in your account:
   - Custom Records
   - Custom Transactions.
   - Advanced PDF/HTML Templates
   - Client SuiteScript
   - Server SuiteScript
   - Web Services

   To enable features, see the help topic Enabling Features.
3. Make sure the Company Information page has a return email address.

   Important: If the **Return Email Address** field is blank, the system will encounter script errors when accessing Electronic Invoicing features.

   a. Go to Setup > Company > Company Information.
   b. In the **Return Email Address** field, enter a valid email address for the company.
   c. Click **Save**.

   Note: If you have a OneWorld account , you must set up the Return Email per subsidiary that uses the Electronic Invoicing SuiteApp.
4. Define the Recipient of E-Document Notifications on the Company Information page.
   a. Go to Setup > Company > Setup Tasks > Company Information.
   b. In the **Recipient of E-Document Notifications** field, enter the email of the user whom you want to receive notifications about e-document processing.
   c. Click **Save**.

   If no notification recipient is defined, the system will send notifications about e-document processes to all active administrators.

Installing the Electronic Invoicing SuiteApp

1. Make sure all prerequisites are met. See Prerequisites for Using Electronic Invoicing.
2. Go to Customization > SuiteBundler > Search & Install Bundles.
3. In the Keywords box, enter the bundle ID or name:
   - Bundle ID: 436209
   - Bundle Name: Electronic Invoicing
4. Click **Search**.
5. Click the link for the Electronic Invoicing SuiteApp.
6. On the Bundle Details page, click **Install**.

The Electronic Invoicing SuiteApp is a managed SuiteApp. When improvements or new features are added to the SuiteApp, your account is automatically updated.

But in sandbox, the update is not automatic. Your administrator must manually update the SuiteApp to get the latest version.

For information about purchasing a license for the Electronic Invoicing SuiteApp, contact your NetSuite account representative.

Considerations When Setting Up Electronic Invoicing

Before you set up an account to use the Electronic Invoicing SuiteApp, it is that you consider the following questions to help you decide on the settings to apply:

For outbound e-document processing:

- To whom should I send my e-documents?
- What e-document templates do I need to use?
- How do I want to send my e-documents to stakeholders?
- Do I want to generate outbound e-documents one by one or in bulk?

For inbound e-document processing:

- From whom should I receive e-documents?
- How do I want to receive the inbound e-documents from vendors or other parties?
- Do I want to convert inbound e-documents into vendor bills, one by one or in bulk?

Be sure to also read Electronic Invoicing Limitations and Best Practices.

Electronic Invoicing Setup Tasks

After installing the Electronic Invoicing SuiteApp, an administrator must accomplish some common setup tasks. As the Electronic Invoicing SuiteApp is composed of two major components, the administrator must also complete specific setup tasks for outbound e-document processing and inbound e-document processing. Some inbound and outbound settings have a common record.

Common Setup Tasks for Outbound and Inbound E-Document Processing:

1. Install the Electronic Invoicing SuiteApp.
2. Define the country for free use (required only if you choose not to purchase a license). See Defining the E-Document Country for Free Use.
3. Create an e-document package and assign it to customer records (for outbound) or vendor records (for inbound). An e-document package record contains inbound and outbound e-document templates, outbound e-document sending methods, and inbound validation plug-in. See Creating E-Document Packages.
4. Create e-document templates. An e-document template record contains templates for both outbound and inbound processing. See Creating E-Document Templates.

Setup Tasks for Outbound Processing:

1. Create e-document sending methods. See Creating E-Document Sending Methods.
2. (Optional) Select a designated e-document sender. See Selecting a Designated E-Document Sender.

Setup Tasks for Inbound Processing:

1. (Optional) Create an inbound e-document validation plug-in. See Inbound Validation Plug-ins.
2. (Optional) Create an XSD file for automatic template selection. See the help topic Understanding XSD in Inbound E-Document Templates.
3. Enable the Email Capture Plug-in. See Electronic Invoicing Inbound Email Capture.
4. (Optional) Use SOAP web services to receive inbound e-documents. See Using SOAP Web Services for Inbound Processing
5. Update item records.
6. Update vendor records.
7. Define the schedule for automatic bulk conversion. See Deploying Automatic Bulk Conversion Script for Inbound E-Documents.

Defining the E-Document Country for Free Use

The Electronic Invoicing SuiteApp can be used for free if you intend to send e-documents to only one country.

To set up your account for free use of the Electronic Invoicing SuiteApp, you must go to the Company Information page and define the country to which you will be sending e-documents.

This is the first step in setting up your account so that you can generate e-documents and send them to customers or vendors. You can send e-documents only to those customers whose country billing addresses match the e-document country for free use.

For Single Instance accounts that have NetSuite SuiteApp License Client installed, it is not required to define the country. The country selected in the **E-Document Country for Free Use** field is not applicable for Single Instance accounts. Single Instance accounts have single business in only the country where their business operates.

Note: For single instance accounts that have NetSuite SuiteApps License Client installed, you need not define the country. The **E-Document Country for Free Use** field is disabled and populates the company's country because these accounts have a single business in only the country where their business operates. The Electronic Invoicing SuiteApp can be used without any license for this scenario.

Defining E-document Country for Free Use

You can follow the following steps to select a country for free use.

To define the e-document country for free use:

1. Go to Setup > Company > Company Information.
2. In the **E-Document Country for Free Use** field, select the country to which you want to send e-documents. Only countries with subsidiaries are available in the list. This setting applies to all subsidiaries of the parent company.

   Note: If you have a license for multi-country use of the Electronic Invoicing SuiteApp, there is no requirement to select a free country. You can use it for all countries without any restriction. Hence, the field will be unavailable.
3. Click **Save**.

After defining the e-document country for free use, you can create e-document packages. See Creating E-Document Packages.

Note: If the license is inactive in One World Account, E-document Free Country Use field will be available. However, as it gets active, the field will be unavailable. In single instance accounts, the E-document Free Country field is always disabled irrespective of license status.

Advanced PDF/HTML Template

You can create a new Advanced PDF/HTML Templates or customize an existing Advanced PDF/HTML template. Along with it, you can also add an element for displaying the QR Code in a generated E-Document. From the code snippet below, refer to "custbody_qr_string" which is the field ID that holds the QR string such as base 64 string value after successful generation of QR. You can use the free marker syntax with "if condition" to render only if the value is present in that field.

To create an Advanced PDF/HTML template, go to Customization>Forms>Advanced PDF/HTML Templates.

```
<#if record.custbody_qr_string?has_content>
  <table style="width: 100%;">
    <tr>
      <td align="center">
        <barcode codetype="qrcode" showtext="false" height="150" width="150" value="${record.custbody_qr_string}"/>
      </td>
    </tr>
  </table>
</#if>
```

Setting Advanced PDF/HTML Template

1. Go to Setup > Company > Subsidiaries for OW account, and Setup > Company > Company Information for SI account.
2. The field “Advanced PDF/HTML Template“ helps choose a template from the list of advanced PDF/HTML templates. You can select one of the advanced PDF/HTML templates to generate the E-Document in the particular format.

   If no template is chosen in subsidiary record/company information, the generated e-document will take the template selected on the "Print Template" field in transaction form. In case of standard forms, it takes the default print template.

Note: The selection of the template in subsidiary record/company information page is only applicable for invoice and credit memo.

Granting Access Permission to the E-Documents Portlet

Roles that have default permissions to access and perform e-document processes, will be able to add and display the e-documents portlet on the Home page. These roles include:

- AR Clerk
- AP Clerk
- Accountant
- Administrator
- Bookkeeper
- CFO
- Custom Role for Accounting Center.

To grant roles the access permission to the e-documents portlet:

1. Go to Customizations > Scripting > Script Deployments.
2. On the Script Deployments page, click the Filters plus sign (+) to display the filters. In the **Type** filter, select **Portlet** from the dropdown list.
3. Click the Edit link of the E-Document Dashboard PT script.
4. On the script deployment page of the portlet, go to the **Audience** subtab. In the **Roles** dropdown list, select the roles that you want to grant permission to access the e-documents portlet.
5. Click **Save**.

Understanding E-Documents and E-Document Packages

The following topics describe e-documents and e-document packages according to how they are used by NetSuite and the Electronic Invoicing SuiteApp.

E-Document

An e-document is a NetSuite transaction represented as an XML or JSON document generated according to a specified standard.

E-documents are transmitted to interested external parties, such as customers, vendors, or tax agencies. E-documents can be downloaded or shared through email, or transmitted through a web service or other custom method.

Some countries legally recognize these XML or JSON files as official documents, whereas other countries recognize the printed transactions as the officially accepted documents.

For information about currently supported transactions, see Electronic Invoicing Limitations and Best Practices.

E-Document Package

An e-document package defines a set of formats used for various transactions to ensure clear communication between all stakeholders of a specific business process, such as vendors communicating with customers, and vendors communicating with tax authorities.

For example, the PEPPOL e-document package defines all possible transactions between vendors and government agencies (sales orders, invoices, credit memos, return authorizations, and others).

An e-document package specifies the following:

- the file format and template
- the communication medium for transmitting the e-documents
- the process necessary for ensuring authenticity and privacy of the information

An e-document package is usually defined by a tax authority or an industry governing body. Some e-document packages are defined by independent entities such as PEPPOL and ISO.

Creating E-Document Packages

An administrator must first create e-document packages so that users can assign them to customer and vendor records.

An e-document package defines the e-document templates and sending methods to be used for the customer's or vendor's transactions.

The Electronic Invoicing SuiteApp includes a default e-document package record to which you can associate e-document templates and sending methods. The default e-document package cannot be edited or deleted.

To create an e-document package:

1. Go to Setup > E-Documents > E-Document Package > New.
2. In the **Name** field, enter a name for the e-document package.
3. In the **Inbound Validation Plugin** field, select a validation plug-in.

   for more information about the validation plug-in, see Inbound Validation Plug-ins.
4. (Optional) In the **Description** field, enter text that describes this e-document package.
5. Click **Save**.

You can now assign e-document templates and sending methods to the e-document package. For more information, see Creating E-Document Templates and Creating E-Document Sending Methods.

Multi-subsidiary Support in the Outbound Process

If you are using OneWorld accounts and have multi-subsidiary customer enabled, outbound e-documents templates and sending methods can be associated with subsidiaries as preferred or necessary. To do this, the administrator must specify subsidiaries in the **Subsidiary** field on an outbound e-document template or sending method record. The transactions of the selected subsidiaries will use the outbound template and sending method. For more information, see Creating E-Document Templates and Creating an E-Document Sending Method Record.

If only one outbound template or one sending method is associated with a customer or vendor, you can set automatic selection of that template and sending method for the transactions of the customer or vendor. To do this, edit a customer or vendor record and go to the E-Document subtab, then check the **Template and Sending Method Auto-selection** box and save the record. The template and sending method will be automatically selected as the default value of the E-Document Template or E-Document Sending Method field on the transaction records of the customer or vendor.

Creating E-Document Templates

In outbound processing, the e-document template maps what data in a NetSuite transaction record will populate which elements in the XML or JSON file that will be generated and sent to customers or vendors.

In inbound processing, the opposite is implemented. The e-document template maps what elements in the received XML file will populate which data fields in the NetSuite transaction record that will be created from the XML file.

Both outbound and inbound e-document templates can be created on the E-Document Templates record. In this record, you can define the name of the e-document template, the e-document package that the template will be applied to, the applicable transaction types for the template, and the template content.

The template content is different for an outbound and inbound transaction. An outbound e-document template should have content in XML or JSON freemarker template, whereas an inbound e-document template will have content as JSON objects for the field mapping.

An e-document template can be used for both outbound and inbound e-documents, given that the right transaction types are selected and the template content for both outbound and inbound are defined. You can create or customize e-document templates for each country you do business in or for certain industries, according to specified standards.

Use FreeMarker to create the template content for both outbound and inbound e-documents. For more information about scriptable templates and FreeMarker, see the following topics:

- Scriptable Templates
- FreeMarker Data Model
- FreeMarker Syntax

Or, you can customize the sample outbound and inbound e-document templates included with the Electronic Invoicing SuiteApp.

The sample outbound template is based on the Brazil NF-e standard. It can be downloaded from the File Cabinet in the Sample Templates folder of the SuiteApp in Documents > File Cabinet > SuiteBundles > Bundle 436209. The following table lists the sample e-document templates and details.

Filename	Description
customer_payment_sample.xml	sample outbound e-document template with tags for customer payment
inbound_template_expenses_items_sample.txt	sample inbound e-document template with tags for items and expenses
inbound_template_sample.txt	sample inbound e-document template with tags for items only
NFe_sample.xml	sample outbound e-document template for Brazil
itemfulfillment_sample.xml	sample outbound e-document template with tags for customer and company information, includes PO number and items
vendorbill_sample.xml	sample outbound e-document template with tags for vendor and company information, includes items and expenses

To create an e-document template:

1. Go to Setup > E-Documents > E-Document Templates > New.
2. In the **Name** field, enter a name for the template.
3. (Optional) In the **Description** field, enter text that describes this template.
4. In the **E-Document Package** field, select the e-document package that this template will be associated with.

   For more information, see Creating E-Document Packages.

   > **Note:** If the template is for outbound, the e-document package that this template will be included in, must be assigned to corresponding customers or vendors.

5. (Optional) Select a plug-in implementation in the **Digital Signature Plugin Implementation** field. The selected plug-in enables you to include a Digital Signature to the e-documents that will be generated using the template. For more information, see Creating a Digital Signature Plug-in Implementation for E-Documents.
6. (Optional) Select a plug-in implementation in the **Outbound Validation Plug-in Implementation** field. The selected plug-in implementation enables you to validate the outbound e-document generation process. For more information, see Creating an Outbound Validation Plug-in Implementation for E-Documents.
7. (Optional) Select a plug-in implementation in the **Custom Data Source Plugin Implementation** field. The selected plug-in enables you to include a custom data source in the template for adding more field values to e-documents that will be generated using the template. For more information, see Creating a Custom Plug-in Implementation for E-Document Custom Data Source.

8. In the **Transaction Type** field, select one or more transaction types for which this template will be used. To select multiple transaction types, press and hold the **Ctrl** key while selecting the transaction types.

   Note: Make sure that you select the right transaction type that match the template you are creating. If you are creating an outbound template, you must select transaction types that are applicable to outbound processing. See Transactions and Processes Supported by the Electronic Invoicing SuiteApp to know which transaction types you can select.

   Selecting a transaction type for either outbound or inbound will make its corresponding Template Content field required. For example, if you select the outbound transaction type Invoice, the **Template for Outbound E-Documents** field will be required. On the other hand, if you select Bill, which is an inbound transaction type, the JSON **Field Mapping for Inbound E-Documents** field will be required.

   The selected transaction types cannot be modified after the template has been used in a transaction. You must remove the e-document template from the transaction before you can modify this field.

9. (For an outbound template) In the **Subsidiary** field, select the subsidiaries that you want to associate with this template. To select multiple subsidiaries, press and hold the **Ctrl** key while selecting the subsidiaries.

   If only this template is associated with a subsidiary, the supported transactions of that subsidiary will display this template on the E-Document Template field on the E-Document subtab. For more information, see Multi-subsidiary Support in the Outbound Process.

10. In the **Restrict Editing of Transactions with E-document Status** field, select any of the following e-document statuses:

    - Sent
    - Sending
    - Ready for Sending
    - Certification in Progress

    To select multiple e-document statuses, press and hold the **Ctrl** key while selecting the statuses. Transactions with the selected e-document status will be locked for editing if this template is associated with them. Those transactions will display a banner message indicating that the transaction cannot be edited. Editing is only locked on the user interface, transactions can still be edited through script.

11. Under the Template Content section, do the following:

    - For outbound processing, select XML or JSON format from the **Content Type** field of the e-document for which the template is generated.
    - If you selected transaction types for outbound processing, enter the XML or JSON content of the outbound e-document template in the **Template for Outbound E-Documents** field.
    - If you selected transaction types for inbound processing, enter JSON content in the **Field Mapping for Inbound E-Documents** field. For more information about the inbound e-document template in JSON format, see Understanding Inbound E-Document Templates in JSON Format.

      If the template record you are creating is for both outbound and inbound transaction types, make sure that you enter content in both Template for Outbound E-Documents, and Field Mapping for Inbound E-Documents fields. Otherwise, error messages will be displayed when you save the template record.

12. Create an XSD file to enable the system to automatically assign the right template to a received XML file after validating required tags and attributes in the XML document. Then, upload the XSD file you created to the File Cabinet. For more information and a sample of the XSD file, refer to Understanding XSD in Inbound E-Document Templates.
13. Return to the E-Documents Templates page, and under Template Content, select the XSD file you created from the **Inbound XSD File** dropdown field.
14. Create an XSD file to enable the system to validate required tags and attributes in the generated XML e-document. Then, upload the XSD file you created to the File Cabinet. For more information and a sample of the XSD file, refer to Understanding XSD in Outbound E-Document Templates.
15. Return to the E-Documents Templates page, and under Template Content, select the XSD file you created from the **Outbound XSD File** dropdown field.
16. Enter the folder ID or folder path of a File Cabinet in the **XSD Folder** field. The folder contains the XSD files imported by the outbound XSD file.
17. (Optional) Add path and regex validation expressions to use for validating the template.

    There is no need to add `node()` at the end to get the value.

    See XPath and Regex Examples for E-Document Templates.
18. Click **Save**.

After creating the e-document template, you can include it in an e-document package, which is then assigned to a customer or vendor. See Creating E-Document Packages

XPath and Regex Examples for E-Document Templates

The examples provided in the table can be used to validate the following sample template:

```
<document>
<buyer>Abuyer</buyer>
<amount>100</amount>
<items>
<item><id>1</id><name>Mouse</name></item>
<item><id>2</id><name>Keyboard</name></item>
<item><id>3.0</id><name>Monitor</name></item>
<count>3.0</count>
</items>
</document>
```

The following table contains sample xpath and regex validation expressions:

Xpath	Regex validation expression	Description	Values for validation	Validation result
"document/buyer"	/^.+$/	The value cannot be blank	Abuyer	Pass
"document/amount"	/^\d+\.\d{0,4}$/	The value is a decimal with up to 4 decimal places	100	Fail
"document/items/item/id"	/^\d+$/	The value is a whole number	Each value will be validated	
			1	Pass
			2	Pass
			3.0	Fail
"document/items/count"	/^\d+(\.*\d{0,2}){1}$/	The value is a whole number or a decimal	3.0	Pass

Xpath	Regex validation expression	Description	Values for validation	Validation result
		number with up to 2 decimal places		

For more information about xpath and regex validations, go to the w3schools website:

- w3schools xpath tutorial
- w3schools regex tutorial

PEPPOL Template

Electronic Invoicing SuiteApp supports PEPPOL standard templates. PEPPOL is an extremely secure international network that enables your company to exchange business critical electronic documents with users who have registered as a part of the PEPPOL network. The use of PEPPOL templates is governed by a multi-lateral agreement structure owned and maintained by OpenPeppol. PEPPOL enables trading partners to exchange standards based electronic documents over the PEPPOL network.

NetSuite suggests you to use the PEPPOL templates along with the Custom Data Source Plug-in implementation to generate the e-document templates correctly. Also, you can customize the template to get the generated e-document template in the required form.

The following table has the names, location of the templates and custom source data plug-in implementations for using the PEPPOL standard templates:

Names	Template Name	Template Location	Plug-in Implementation Name
Invoice Template	Invoice_Peppol_Template_ Master.txt	Bundle 436209/Sample Templates	
Generic Invoice Template	Invoice_Peppol_Template_ Generic.txt	Bundle 436209/Sample Templates	
Credit Memo Template	CreditMemo_Peppol_Template_ Master.txt	Bundle 116079/Sample Templates	
Generic Credit Memo Template	CreditMemo_Peppol_Template_ Generic.txt	Bundle 116079/Sample Templates	
Custom Data Source Plugin	pl_custom_data_source_peppol.js	Bundle 436209/src/comp/pl	PEPPOL Outbound Custom Data Source

Note: To use these templates, you must perform the following actions:

- Create an e-document template by adding the content of the template in the **Templates for Outbound E-Documents** field, in the Create E-Document Templates page.
- Select the PEPPOL Outbound Custom Data Source plug-in implementation from the **Custom Data Source Plug-in Implementation** drop down list.

The following table gives an overview of all the elements that are used in the and Generic PEPPOL templates for Invoice and Credit Memo transaction types . The elements in the following table use hard

coded values or values from the transactions and its related to Subsidiary, Customer and other records. The remaining elements are sourced from custom data source plug-in implementation.

SL No.	ID	Field ID	Technical Name of Field	Transaction Type	NetSuite Field Mapping Logic Used for Templates	NetSuite Field Mapping Logic Used for Generic Templates
1	ubl:Invoice			Invoice	This is the root element of the invoice's PEPPOL template. It has a default value.	This is the root element of the invoice's PEPPOL template. It has a default value.
2	ubl:CreditNote			Credit Memo	This is the root element of the credit memo's PEPPOL template. It has a default value.	This is the root element of credit memo's PEPPOL template. It has a default value.
3	cbc:CustomizationID			Both	This element identifies the specifications of the rules for semantic content, cardinalities, and business to which the data in the instance document conforms to. This element has a default value.	This element identifies the specifications of a set of rules for semantic content, cardinalities, and business to which the data in the instance document conforms. This element has a default value.
4	cbc:ProfileID			Both	This element identifies the context of the business process related to the transaction and lets the buyer process the invoice accurately. This element has a default value.	This element identifies the context of the business process related to the transaction and lets the buyer process the invoice accurately. This element has a default value.
5	cbc:ID	transaction.tranid	Invoice Number or Credit Number	Both	This element displays a unique number that identifies if the transaction is an Invoice or Credit Memo. This value is mapped with the entry number of the transaction.	This element displays a unique number that identifies if the transaction is an Invoice or Credit Memo. This value is mapped with the **entry number** of the transaction.
6	cbc: IssueDate	transaction.trandate	Date	Both	This element displays the issue date value present in the **Date** field of the transaction. The date format is YYYY-MM-DD in a template.	This element displays the issue date value present in the **Date** field of the transaction. The date format is YYYY-MM-DD in a template.
7	cbc: DueDate	transaction.duedate	Due Date	Invoice	This element displays the payment due date value present in the **Due Date** field of an Invoice. The date format is YYYY-MM-DD in a template.	This element displays the payment due date value present in the **Due Date** field of an Invoice. The date format is YYYY-MM-DD in a template.
8	cbc:InvoiceTypeCode		Custom Field	Invoice	This element displays a code specifying the invoice type. This has a default assigned value of 380. You must map the value with the required custom field in the SuiteApp. The following list has information about the Invoice list and the Invoice Type Codes (UNCL1001): ■ Commercial Invoice – 380 ■ Consignment Invoice – 395 ■ Debit Note for Financial Adjustments – 84 ■ Debit Note for Goods or Services – 80 ■ Debit Note – 383 ■ Factored Invoice – 393 ■ Forwarder's Invoice – 623 ■ Insurer's Invoice – 575 ■ Metered Services Invoice – 82 ■ Prepayment Invoice – 386 You must map the value with the required custom field in the SuiteApp.	This element displays a code specifying the invoice type. This has a default assigned value of 380. You must map the value with the required custom field in the SuiteApp. The following list has information about the Invoice list and the Invoice Type Codes (UNCL1001): ■ Commercial Invoice – 380 ■ Consignment Invoice – 395 ■ Debit Note for Financial Adjustments – 84 ■ Debit Note for Goods or Services – 80 ■ Debit Note – 383 ■ Factored Invoice – 393 ■ Forwarder's Invoice – 623 ■ Insurer's Invoice – 575 ■ Metered Services Invoice – 82 ■ Prepayment Invoice – 386
9	cbc:CreditNoteTypeCode		Custom Field	Credit Memo	This element displays a code specifying the credit memo type. This has a default assigned value of 381. You can map the value with the required custom field in the SuiteApp. The following list has the information about Credit Memo list and the Credit Memo Type Codes (UNCL1001): ■ Credit Note for Foods or Services – 81 ■ Credit Note for Financial Adjustments – 83	This element displays a code specifying the credit memo type. This has a default assigned value of 381. You can map the value with the required custom field in the SuiteApp. The following list has the information about Credit Memo list and the Credit Memo Type Codes (UNCL1001): ■ Credit Note for Foods or Services – 81 ■ Credit Note for Financial Adjustments – 83 ■ Credit Note – 381

					▪ Credit Note – 381 ▪ Factored Credit Note – 396 ▪ Forwarder's Credit Note – 532 You can map the value with the required custom field in the SuiteApp.	▪ Factored Credit Note – 396 ▪ Forwarder's Credit Note – 532
10	cbc: Note	transaction.memo	Memo or Custom Field	Both	This element displays a note with information about any change in the invoice or credit memo. This element is displayed only if there is a value in the **Memo** field of a transaction. In some SuiteApps, this value can be a combination of the value from the **Memo** field and the value from the custom field on a Subsidiary record. You can map the value with the required custom field in the SuiteApp.	This element displays a note with information about any change in the invoice or credit memo. The note is retrieved from the value in the **Memo** field of a transaction. In some SuiteApps, this value can be a combination of the value from the **Memo** field and the value from the custom field on a subsidiary record. You can map the value with the required custom field in the SuiteApp.
11	cbc: Tax Point Date	transaction. taxpointdate	Tax Point Date	Both	This element displays the **Tax Point Date** field value on the Tax Details tab of a transaction. This field value must not be the same as the value of the **Date** field. It is available only in accounts with SuiteTax enabled. This element is displayed only if there is a value in the **Tax Point Date** field of the Transaction. The Tax Point Date format is YYYY-MM-DD in a template. The Tax Point Date is the date when the VAT becomes accountable for the seller and buyer. This field is required and acceptable only if this date is different from the transaction's issue date, according to the VAT directive.	This element displays the **Tax Point Date** field value on the Tax Details tab of a transaction. This field value must not be the same as the value of the **Date** field. It is available only in accounts with SuiteTax enabled. This element is displayed only if there is a value in the **Tax Point Date** field of the Transaction. The Tax Point Date format is YYYY-MM-DD in a template. The Tax Point Date is the date when the VAT becomes accountable for the seller and buyer. This field is required and acceptable only if this date is different from the transaction's issue date, according to the VAT directive.
12	cbc:Document CurrencyCode	transaction. currency.symbol	Currency	Both	This element displays the ISO Code of the currency used in the transaction, if Multi-Currency feature is enabled in an account. The value is not displayed if the Multi-Currency feature is not enabled in an account.	This element displays the ISO Code of the currency used in the transaction, if Multi-Currency feature is enabled in an account. The value is not displayed if the Multi-Currency feature is not enabled in an account.
13	cbc:TaxCurrencyCode	custom.subCurrencyISOCode Refer Custom Data Source (CDS) Plug-in Implementation	Currency — In Subsidiary Record	Both	This element displays the ISO Code used by the subsidiary of the transaction if the currency is not same for transaction and the subsidiary. The value of this element is retrieved from the custom data source plug-in used in the template. TaxCurrencyCode is used for VAT accounting and reporting, as accepted or required in the country of the seller.	This element displays the ISO code of the Currency used by the subsidiary in a transaction in One World (OW)Accounts. In Single Instance (SI) accounts the value of the **Currency** field in Company Information Page is used. The ISO code is displayed when the value of the currency and the subsidiary are not the same for the transaction and Multi Currency feature is enabled in the account. The value of this element is retrieved from the custom data source plug-in used in the template. TaxCurrencyCode is used for VAT accounting and reporting purposes as accepted or required in the country of the Seller.
14	cbc:AccountingCost		Custom Field	Both	This element displays the details of the booking data related to the buyer's financial accounts. You must map this field value with the required custom field in the SuiteApp.	The template does not support this element.
15	cbc:BuyerReference	transaction. otherrefnum	PO #	Both	This element displays the identifier or code assigned by the buyer for internal routing. The **PO#** field is used to identify the buyer and the order details. This element is displayed only if the **PO#** field has a value in a transaction. The value can be the name of the person ordering the products, employee number, or an identification code for a buyer, department, or group. You can map the value of this field with the required custom field in the SuiteApp.	This element displays the identifier or code assigned by the buyer for internal routing. The **PO#** field is used to identify the buyer and the order details. This element is displayed only if the **PO#** field has a value in a transaction. The value can be the name of the person ordering the products, employee number, or an identification code for a buyer, department, or group. You can map the value of this field with the required custom field in the SuiteApp.
16	cac:InvoicePeriod			Invoice	This is a parent element containing a group of business terms that provide information about the invoice period, also known as the	This is a parent element containing a group of business terms that provide information about the invoice period, also known as the delivery

					delivery period. If you use this group, you must also use the invoice period start date, end date, or both. This parent element is not displayed if any of the child elements do not have a value.	period. If you use this group, you must also use the invoice period start date, end date, or both. This element is displayed only if the value of start date, end date, or both is available.
17	cbc: Start Date	transaction. startdate	Start Date	Invoice	This element displays the start date of an invoice period for a transaction. This element is displayed along with a value on the generated e-document only if the value is entered in the **Start Date** field of the transaction. You can map the value with the required custom field in the SuiteApp.	This element displays the start date of an invoice period for a transaction. This element is displayed along with a value on the generated e-document only if the value is entered in the **Start Date** field of the transaction.
18	cbc: End Date	transaction. enddate	End Date	Invoice	This element displays the end date of an invoice period for a transaction. This element is displayed along with a value on the generated E-Document only if the value is entered in the **End Date** field of the transaction. You can map the value with the required custom field in the SuiteApp.	This element displays the end date of an invoice period for a transaction. This element is displayed along with a value on the generated E-Document only if the value is entered in the **End Date** field of the transaction.
19	cbc:DescriptionCode		Custom Field	Both	This element displays the code of the date when VAT becomes accountable for seller and buyer. For example: ■ 3 - Invoice document issue date, time ■ 35 - Delivery date, time ■ 42 - Paid to date You can map the value with the required custom field in your account.	The template does not support this element.
20	cac:OrderReference			Both	This is a parent element with details of the order and sales order references.	This is a parent element with details of the order and sales order references.
21	cbc:ID	transaction. otherrefnum customer.account number customer.entityid	PO# Account Number Customer ID	Both	This element displays one the following identifiers for a referenced purchase order allotted by the buyer: ■ PO# ■ Account Number – The account number is displayed if the PO# value is missing. ■ Customer ID – The customer ID is displayed if the account number is missing. The element represents an identifier for a referenced purchase order issued by the Buyer.	This element displays one the following identifiers for a referenced purchase order allotted by the buyer: ■ PO# ■ Account Number – The account number is displayed if the PO# value is missing. ■ Customer ID – The customer ID is displayed if the account number is missing. The element represents an identifier for a referenced purchase order issued by the Buyer.
22	cbc:SalesOrderID	transaction. createdfrom	Created From	Invoice	This element displays an identifier for a referenced sales order allotted by the seller. It has the entry number of the sales order transaction only if an invoice is created from a sales order. The entry number is retrieved from the **Created From** field in the invoice. If the invoice is not created from a sales order, then the element is not printed.	This element displays an identifier for a referenced sales order allotted by the seller. It has the entry number of the sales order transaction only if an invoice is created from a sales order. The entry number is retrieved from the **Created From** field in the invoice. If the invoice is not created from a sales order, then the element is not printed.
23	cac:BillingReference			Credit Memo	This is a parent element containing a group of business terms with information about previous invoices. In the template, the element has no value and has child elements that can have values if required.	This is a parent element containing a group of business terms with information about previous invoices. In the template, the element has no value and has child elements that can have values if required.
24	cac:InvoiceDocumentReference			Credit Memo	This parent element has no value in the template but contains child elements identifying an invoice.	This parent element has no value in the template but contains child elements identifying an invoice. This element has no value in the template.
25	cbc:ID	transaction. createdfrom	Created From	Credit Memo	This element displays a value identifying an invoice that was previously sent by the seller. If a credit memo has an invoice applied, then the value of the invoice entry is displayed in the **Created From** field.	This element displays a value identifying an invoice that was previously sent by the seller. If a credit memo has an invoice applied, then the value of the invoice entry is displayed in the **Created From** field.

26	cbc: Issue Date	transaction.createdfrom.trandate	Date – Invoice's **Date** field	Credit Memo	This element displays the date when the last invoice was issued. If a credit memo has an invoice applied to it, then the value of the invoice entry is displayed in the **Created From** field. The **Date** field value represents the issue date of that invoice.	This element displays the date when the last invoice was issued. If a credit memo has an invoice applied to it, then the value of the invoice entry is displayed in the **Created From** field. The **Date** field value represents the issue date of that invoice.
27	cac:DespatchDocumentReference			Both	This parent element provides reference to the dispatch advice. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
28	cbc:ID		Custom Field	Both	This element refers to an identifier of a referenced dispatch advice. You can map the value of this key with the preferred custom field in your account.	The template does not support this element.
29	cac:ReceiptDocumentReference			Both	This parent element has reference to the receipt advice. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
30	cbc:ID		Custom Field	Both	This element displays an identifier of a referenced receiving advice. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
31	cac:OriginatorDocumentReference			Both	This parent element contains a tender or lot reference details. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
32	cbc:ID		Custom Field	Both	This element displays the call for tender or lot of the invoice ID. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
33	cac:ContractDocumentReference			Both	This parent element provides details of any existing contract associated with a transaction. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
34	cbc:ID		Custom Field	Both	This element has a value that identifies a contract. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
35	cac:AdditionalDocumentReference			Both	This parent element has no value, but it has child elements with values associated to it in the template. It has information about group of business terms and details, with additional supporting documents having terms of the invoice. The supporting documents include: ▪ Reference to a document number known to the receiver ▪ External documents referenced by a URL ▪ Embedded documents ▪ Base64 encoded documents like time report	The template does not support this element.
36	cbc:ID		Project	Both	This element displays the identifier of an object based on the invoice provided by the seller or a value used for identifying a supporting document. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
37	@schemeID		Custom value from list	Both	This element must have a value displaying the identifier of the identification scheme of an invoiced object. It is used inside the	The template does not support this element.

					cbc:ID element. The value of this element must be from the list. You can map the value with the required custom field in the SuiteApp.	
38	cbc:DocumentTypeCode		Custom Field	Both	This element displays the code value 50 if there is a project specified in the header of the credit memo. If the credit memo has an applied invoice or it does not display any content if conditions are not met, then the code value is 130 instead of 50. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
39	cbc:DocumentDescription		Custom Field	Both	This element has a value that describes supporting documents such as time sheets and usage reports. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
40	cac:Attachment			Both	This parent element has information about attached documents with the transaction. The element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
41	cbc:EmbeddedDocumentBinaryObject		Custom Field	Both	This element has a value referencing any attached document embedded as a binary object (Base64) or sent together with the invoice. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
42	@mimeCode		Custom Value	Both	This is a value used with the element cbc:EmbeddedDocumentBinaryObject that represents the mime code of an attached document. For example: text/csv You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
43	@fileName		Custom Value	Both	This is a value used with the element cbc:EmbeddedDocumentBinaryObject that represents the file name of an attached document. For example: abc.csv You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
44	cac:External Reference			Both	This is a parent element providing details of the external document's reference such as its location. The element has no value, but it has child elements with values associated to it in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
45	cbc:URI		Custom Field	Both	The value of this element represents the Uniform Resource Locator (URL) that identifies the location of the external document. This locates the resource and its primary access mechanism, such as http:// or ftp://. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
46	cac:ProjectReference			Invoice	This parent element has information about any project associated with an invoice. This element has no value, but it has child elements with values associated to it in the template. This element is not displayed if its child element do not have any value. You can map the value with the required custom field in the SuiteApp.	This parent element has information about any project associated with an invoice. This element has no value, but it has child elements with value associated to it in the template. This element is not displayed, if the **Project** field has no value. You can map the value with the required custom field in the SuiteApp.
47	cbc:ID	transaction.job	Project	Invoice	This element indicates the ID of the project associated with an invoice. If the **Project**	This element indicates the ID of the project associated with an invoice. If the **Project** field has

					field has no value in the transaction, the element is not printed.	no value in the transaction, the element is not printed.
48	cac:AccountingSupplierParty			Both	This is a parent element that provides information about the seller using various child elements. This element has no value, but it has child elements with values associated to it in the template.	This is a parent element that provides information about the seller using various child elements. This element has no value, but it has child elements with values associated to it in the template
49	cac:Party			Both	This is a parent element and a child element of cac:AccountingSupplierParty, having information about the seller like electronic address. This element has no value, but it has child elements with values associated to it in the template.	This is a parent element and a child element of cac:AccountingSupplierParty, having information about the seller like electronic address. This element has no value, but it has child elements with values associated to it in the template.
50	cbc:EndpointID		Vat Registration No Subisidiary Tax Reg Number	Both	This element identifies the seller's electronic address to which the application level response to the invoice can be delivered. The element uses the custom data source plug-in to get one of the following values: ■ Value of the **VAT Registration No** field from the subsidiary in legacy tax accounts or ■ Value of the **Subsidiary Tax Reg Number** field from the transaction for SuiteTax accounts. You can map the value with the required custom field in the SuiteApp.	This element identifies the seller's electronic address to which the application level response to the invoice can be delivered. The element uses the Custom Data Source plugin to get one of the following values: ■ Value of the **VAT Registration No** field from the subsidiary in legacy tax accounts or ■ Value of the **Subsidiary Tax Reg Number** field from the transaction for SuiteTax accounts. You can map the value with the required custom field in the SuiteApp.
51	@schemeID		Custom Value	Both	This element is used with the element cbc:EndpointID and identifies the scheme ID of the seller's electronic address. You can find the scheme ID value from Scheme ID List. You can map the value with the required custom field in the SuiteApp.	This element is used with the element cbc:EndpointID and identifies the scheme ID of the seller's electronic address. You can find the scheme ID value from Scheme ID List. You can map the value with the required custom field in the SuiteApp.
52	cac:PartyIdentification			Both	This parent element identifies the seller or the seller's unique banking reference ID provided by the bank. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
53	cbc:ID		Custom Field	Both	This parent element identifies the seller or the seller's unique banking reference ID provided by the bank. ICD code list is used for the seller's ID. SEPA is the code used for SEPA bank's assigned creditor reference. For the buyer to identify a supplier automatically, the following identifiers are available: ■ BT-29 – Seller identifier ■ BT-30 – Seller legal registration identifier ■ BT-31 – Seller VAT identifier You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
54	@schemeID		Custom Value	Both	This element is used with the element cbc:ID to identify the scheme of the seller Identifier. For example, for a bank assigned credit identifier BT-90, the seller ID must be SEPA.	The template does not support this element.
55	cac:PartyName			Both	This parent element displays the seller's information such as name and address. This element has no value, but it has child elements with values associated to it in the template. This element is not displayed if its child elements do not have values.	This parent element displays the seller's information such as name and address. This element has no value in the template, but it has child elements with values associated with them. This element is not displayed If the **Addressee** field value of the subsidiary's main address or **Name** field value of the subsidiary is not present.
56	cbc:Name	transaction.subsidiary.name custom.addressee	Addressee (Subsidiary's main address's addressee name) Name (Subsidiary name)	Both	This element displays the business name, another name by which the Seller is known, other than the Seller name. It contains a value in the **Addressee** field of the Subsidiary's main address. If the field has no value, then this element prints the Subsidiary name without the full hierarchy. This	This element displays the business name, another name by which the Seller is known, other than the Seller name. It contains a value in the **Addressee** field of the Subsidiary's main address retrieved from custom data source plug-in. If the **Addressee** field has no value, then this element prints the Subsidiary name without the

					element prints the name of the subsidiary without a OneWorld account.	full hierarchy. In Single Instance accounts, the **Name** of the subsidiary to which the transaction is mapped is printed. Example: Parent Company, Arizona Company etc
57	cac:PostalAddress			Both	This is a parent element with information about the seller's address. Required fields of the address must be filled out to comply with the legal requirements. This element has no value, but it has child elements that have values associated to it in the template.	This is a parent element with information about the seller's address. Required fields of the address must be filled out to comply with the legal requirements. This element has no value, but it has child elements that have values associated to it in the template.
58	cbc:StreetName	transaction.subsidiary.address1	Address 1 (Subsidiary's main address's address line 1)	Both	This element displays the main address line of the seller's address. The element has the value of the **Address 1** field of the subsidiary's main address. This element is displayed only if a value is entered in the **Address 1** field on the Subsidiary record; otherwise, the element is not displayed.	This element displays the main address line of the seller's address. The element has the value of the **Address 1** field of the subsidiary's main address. This element is displayed only if a value is entered in the **Address 1** field on the Subsidiary record; otherwise, the element is not displayed.
59	cbc:AdditionalStreetName	transaction.subsidiary.address2	Address 2 (Subsidiary's main address's address line 2)	Both	This element displays another address line with more details about the main address. This element has the value of the **Address 2** field in the subsidiary's main address. This element is displayed only if a value is entered in the **Address 2** field on the Subsidiary record; otherwise, the element is not displayed.	This element displays another address line with more details about the main address. This element has the value of the **Address 2** field in the subsidiary's main address. This element is displayed only if a value is entered in the **Address 2** field on the Subsidiary record; otherwise, the element is not displayed.
60	cbc:CityName	transaction.subsidiary.city	City (Subsidiary's main address's city)	Both	This element displays the common name of the city, town, or village of the seller's location. The element's value is displayed in the **City** field of the subsidiary's main address.	This element displays the common name of the city, town, or village of the seller's location. The element's value is displayed in the **City** field of the subsidiary's main address.
61	cbc:PostalZone	transaction.subsidiary.zip	ZIP(Subsidiary's main address's zip)	Both	This element identifies a group of properties with addresses based on the relevant postal service. This element is displayed only if there is a value in the **Zip** field of the subsidiary's main address.	This element identifies a group of properties with addresses based on the relevant postal service. This element is displayed only if there is a value in the **Zip** field of the subsidiary's main address.
62	cbc:CountrySubentity	Transaction.subsidiary.state	State (Subsidiary's main address's state)	Both	This element identifies the subdivision of the country in the subsidiary's main address. The element is displayed only if there is a value in the **State** field of the subsidiary's main address.	This element identifies the subdivision of the country in the subsidiary's main address. The element is displayed only if there is a value in the **State** field of the subsidiary's main address.
63	cac:AddressLine			Both	This parent element has information about the additional address line in the subsidiary address. This element has no value in the template.	This parent element has information about the additional address line in the subsidiary address. This element has no value in the template.
64	cbc:Line	transaction.subsidiary.address3	ADDRESS 3 (Subsidiary's main address's address line 3)	Both	This element has information about an additional address line with more details about the main address line. This element has the value of the **Address 3** field in the subsidiary's main address. It is displayed only if the **Address 3** field has a value.	This element has information about an additional address line with more details about the main address line. This element has the value of the **Address 3** field in the subsidiary's main address. It is displayed only if the **Address 3** field has a value.
65	cac:Country			Both	This parent element has information about the country associated with the subsidiary's address. This element has no value in the template.	This parent element has information about the country associated with the subsidiary's address. This element has no value in the template.
66	cbc:IdentificationCode	In OW accounts: Country of transaction's subsidiary. In SI account: Country name in Company Information page.	custom.subPrimaryCountry (Refer CDS) custom.countryInCompanyInfoCode(Refer CDS)	Both	This parent element displays the ISO Code of a transaction's country in the subsidiary's main address in OneWorld accounts. Without a OneWorld account, the element displays the ISO Code of the **Country** field on the Company Information page. The values are displayed using the custom data source plug-in implemented in the template.	This element displays the following information: ▪ Value of ISO code of the transaction's subsidiary's **Country** in OW accounts. ▪ Value of the ISO code of the **Country** field in the Company Information page for SI accounts. The values are displayed using the custom data source plug-in implemented in the template.
67	cac:PartyTaxScheme			Both	This parent element with its child element displays the seller's VAT identifier or tax registration details using different transaction fields. This element has no value in the template, but it has child elements that have values associated with them.	This parent element with its child element displays the seller's VAT identifier or tax registration details using different transaction fields. This element has no value in the template, but it has child elements that have values associated with them.

68	cbc:CompanyID	custom.subVatRegNo (Refer CDS)	For Legacy Tax: Employee Identification Number(EIN) on transaction's Subsidiary. For SuiteTax: Subsidiary Tax Reg Number	Both	This element displays the seller's VAT identification number or local identification for tax purpose or reference. This enables a seller to indicate their registered tax status. The element displays the value in the **Employee Identification Number** field of a subsidiary in a transaction in the generated e-document. The value is displayed using the custom data source plug-in implemented in the template.	This element displays the seller's VAT identification number or local identification for tax purpose or reference. This enables a seller to indicate their registered tax status. The element displays the following information about generated e-document: ■ The value of the transaction's subsidiary's **Employee Identification Number** field in Legacy Tax accounts ■ The value of the transaction's **Subsidiary Tax Reg Number** field in Suite Tax accounts. The values are displayed using the custom data source plug-in implemented in the template.
69	cac:TaxScheme				This parent element has a child element cac:PartyTaxScheme that displays the seller's VAT identification or tax registration details using various transaction fields. This element has no value, but it has child elements that have values associated to it in the template.	This parent element has a child element cac:PartyTaxScheme that displays the seller's VAT identification or tax registration details using various transaction fields. This element has no value, but it has child elements that have values associated to it in the template.
70	cbc:ID		Fixed Value: VAT	Both	This element is displayed if the seller is a VAT identifier or TAX Identifier. The default value is VAT.	This element is displayed if the seller is a VAT identifier or TAX Identifier. The default value is VAT.
71	cac:PartyLegalEntity				This parent element has information about the seller's legal details like registration name. This element has no value, but it has child elements that have values associated to it in the template.	This parent element has information about the seller's legal details like registration name. This element has no value, but it has child elements that have values associated to it in the template.
72	cbc:RegistrationName	transaction.subsidiary.legalname	Legal Name	Both	This element displays the seller's registered name as per the following conventions: ■ National Registry of Legal Entities ■ Taxable person ■ Trading as a person or persons The value of the element is retrieved from the subsidiary's **Legal Name** field in a transaction. This element is not displayed if the field has no value.	This element displays the seller's registered name as per the following conventions: ■ National Registry of Legal Entities ■ Taxable person ■ Trading as a person or persons The value of the element is retrieved from the subsidiary's **Legal Name** field in a transaction. This element is not displayed if the field has no value.
73	cbc:CompanyID		Custom Field	Both	This element is used as an identifier allotted by an official registrar. It identifies the seller as legal entity or person. This element does not have any value in the template. You can map the value with the required custom field in the SuiteApp.	This element is used as an identifier allotted by an official registrar. It identifies the seller as legal entity or person. This element does not have any value in the template. You can map the value with the required custom field in the SuiteApp.
74	@schemeID		Custom Value	Both	This element is used with the element cbc:CompanyID to identify the scheme identifier of a seller's legal registration. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element is used with the element cbc:CompanyID to identify the scheme identifier of a seller's legal registration. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
75	cbc:CompanyLegalForm			Both	This parent element has details about additional legal information related to the seller. This element has no value in the template.	The template does not support this element.
76	cac:Contact			Both	This is a parent element and a child of cbc:Party with information about a group of business terms providing contact information of the seller. This element has no value in the template, but it has child elements that have values associated with them. This parent element is not displayed if the child elements do not have values.	This is a parent element and a child of cbc:Party with information about a group of business terms providing contact information of the seller. This element has no value, but it has child elements that have values associated to it in the template. This element is displayed if the transaction has a sales representative.
77	cbc:Name	transaction.salesrep	Sales Rep	Both	This element provides contact details for a legal entity or representative and displays the **Sales Rep** field value in a transaction. The element is displayed only if there is a value in the **Sales Rep** field of the transaction.	This element provides contact details for a legal entity or representative and displays the **Sales Rep** field value in a transaction. The element and the value is not displayed only if there is no value in the **Sales Rep** field of the transaction.

78	cbc:Telephone	transaction.salesrep.phone	Phone	Both	This element displays the phone number of the contact person. It uses the phone number of the sales representative selected in the transaction as the value. This element is displayed only if there is a value in the sales representative's **Phone** field in the employee record.	This element displays the phone number of the contact person. It uses the phone number of the sales representative selected in the transaction as the value. This element and value is not displayed if there is a value in the sales representative's **Phone** field in the employee record.
79	cbc:ElectronicMail	transaction.salesrep.email	Email	Both	This element displays the email address of the contact person and uses the email address of the sales representative selected in the transaction as the value. This element is displayed only if there is value in the sales representative's **Email** field in the employee record.	This element displays the email address of the contact person and uses the email address of the sales representative selected in the transaction as the value. This element is displayed only if there is value in the sales representative's **Email** field in the employee record.
80	cac:AccountingCustomerParty				This parent element contains a group of business terms, with information about the buyer. This element has no value, but it has child elements that have values associated to it in the template.	This parent element contains a group of business terms, with information about the buyer. This element has no value, but it has child elements that have values associated to it in the template.
81	cac:Party			Both	This is a parent element and a child of cac:AccountingCustomerParty, providing information about the buyer. This element has no value, but it has child elements with values associated in the template.	This is a parent element and a child of cac:AccountingCustomerParty, providing information about the buyer. This element has no value, but it has child elements with values associated in the template.
82	cbc:EndpointID	For Legacy Tax: customer.vatregnumber in customer record For SuiteTax: customer.defaulttaxreg in customer record	In Legacy Tax Account: **Tax Reg Number** field in the customer record In SuiteTax account: **Default Tax Reg** field in the customer record	Both	This element displays a value identifying a buyer's electronic address to deliver the invoice. This element is displayed along with the field value based on the following criteria: ▪ A customer record has a value in the **Tax Reg Number** field in Legacy Tax accounts. ▪ A customer record has a value in the **Default Tax Reg** field in SuiteTax accounts. Example: 987654321	This element displays a value identifying a buyer's electronic address to deliver the invoice. This element is displayed along with the field value based on the following criteria: ▪ A customer record has a value in the **Tax Reg Number** field in Legacy Tax accounts. ▪ A customer record has a value in the **Default Tax Reg** field in SuiteTax accounts. Example: 987654321
83	@schemeID		Custom Value	Both	This element is used with the element cbc:EndpointID and has no value in the template. This element identifies the scheme identifier of a buyer's electronic address. Example: 0192 You can map the value with the required custom field in the SuiteApp.	This element is used with the element cbc:EndpointID and has no value in the template. This element identifies the scheme identifier of a buyer's electronic address. Example: 0192 You can map the value with the required custom field in the SuiteApp.
84	cac:PartyIdentification			Both	This is a parent element providing information about the buyer identifier. This element has no value, but it has child elements that have values associated with them.	The template does not support this element.
85	cbc:ID	customer.accountnumber	Account (Customer's transaction)	Both	This element populates the customer identifier. The value of **Account** field on the transaction's customer record is used as value for this element.	The template does not support this element.
86	@schemeID		Custom Value	Both	This element is used with the element cbc:ID to display or populate the scheme ID of the buyer identifier. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
87	cac:PartyName				This parent element displays the buyer's details. This element has no value, but it has child elements that have values associated with it in the template.	This parent element displays the buyer's details. This element has no value, but it has child elements that have values associated with it in the template.
88	cbc:Name	customer.companyname	Company Name	Both	This element displays another name that the buyer is known, other than the Business name. The value in the **Company Name** field on a customer's transaction is displayed in the generated e-document for this element.	This element displays another name that the buyer is known, other than the Business name. The value in the **Company Name** field on a customer's transaction is displayed in the generated e-document for this element.

89	cac:PostalAddress			Both	This parent element has a group of business terms providing information about the buyer's postal address. This element has no value, but it has child elements that have values associated with it in the template.	This parent element has a group of business terms providing information about the buyer's postal address. This element has no value, but it has child elements that have values associated with it in the template.
90	cbc:StreetName	transaction.billaddr1	Address 1 (Transaction 's billing address)	Both	This element displays the main address line of an address. The element retrieves the value of the **Address 1** field from the billing address in a transaction. This element is displayed only if the **Address 1** field has a value.	This element displays the main address line of an address. The element retrieves the value of the **Address 1** field from the billing address in a transaction. This element is displayed only if the **Address 1** field has a value.
91	cbc:AdditionalStreetName	transaction.billaddr2	Address 2 (Transaction 's billing address)	Both	This element displays an additional address line to have more details on the main address. This element retrieves the value of the **Address2** field from the billing address in a transaction. This element is displayed only if the **Address2** field has a value.	This element displays an additional address line to have more details on the main address. This element retrieves the value of the **Address2** field from the billing address in a transaction. This element is displayed only if the **Address2** field has a value.
92	cbc:CityName	transaction.billcity	City (Transaction's billing address)	Both	This element displays the common name of the city, town, or village of the buyer's location. This element is displayed only if the **City** field of the transaction's billing address has a value.	This element displays the common name of the city, town, or village of the buyer's location. This element is displayed only if the **City** field of the transaction's billing address has a value.
93	cbc:PostalZone	transaction.billzip	Zip (Transaction's billing address)	Both	This element identifies a group of properties with address based on the relevant postal service. **Zip** field of the transaction's billing address is used as value for this element. This element is not displayed if this field has no value.	This element identifies a group of properties with address based on the relevant postal service. **Zip** field of the transaction's billing address is used as value for this element. This element is not displayed if this field has no value.
94	cbc:CountrySubentity	transaction.billstate	State (Transaction's billing address)	Both	This element identifies the subdivision of the country. This element is displayed only if the **State** field of the transaction's billing address has a value.	This element identifies the subdivision of the country. This element is displayed only if the **State** field of the transaction's billing address has a value.
95	cac:AddressLine				This parent element shows information about the additional address line in a transaction's billing address. This element has no value, but it has child elements that have values associated with them. This parent element is displayed only if its child elements have values.	This parent element shows information about the additional address line in a transaction's billing address. This element is not displayed if the **Address 3** field of transaction's billing address has no value.
96	cbc:Line	transaction.billaddr3	Address 3 (Transaction 's billing address)	Both	This element displays an additional address line to have more details on the main address line. This element retrieves the value of the **Address 3** field from the billing address in a transaction. This element is only displayed if the **Address 3** field has a value.	This element displays an additional address line to have more details on the main address line. This element retrieves the value of the **Address 3** field from the billing address in a transaction. This element is only displayed if the **Address 3** field has a value.
97	cac:Country				This parent element provides information about the billing address's country in a transaction. This element has no value, but it has child elements that have values associated with it in the template.	This parent element provides information about the billing address's country in a transaction. This element has no value, but it has child elements that have values associated with it in the template.
98	cbc:IdentificationCode	custom.billCountryISOCode	Country (Transaction's billing address)	Both	This element displays a code identifying the billing country of a transaction. The element displays the ISO code of the country. The value of this element is displayed using the custom data source plug-in implemented in the template.	This element displays a code identifying the billing country of a transaction. The element displays the ISO code of the country. The value of this element is displayed using the custom data source plug-in implemented in the template.
99	cac:PartyTaxScheme				This parent element has information about the party VAT identifier. This element has no value, but it has child elements that have values associated to it in the template.	This parent element has information about the party VAT identifier. This element has no value in the template.
100	cbc:CompanyID	For Legacy Tax: customer.vatregnumber For SuiteTax: customer.defaulttaxreg	In Legacy Tax Account: Tax Reg Number In SuiteTax account: Default Tax Reg	Both	This element displays the buyer's VAT identification number. This element is displayed with the field value in a customer record for the following accounts: ▪ **Default Tax Reg Number** in Suite Tax account. ▪ **Tax Reg Number** field in Legacy Tax account. You can map the value with the required custom field in the SuiteApp.	This element displays the buyer's VAT identification number. This element is displayed with the field value in a customer record for the following accounts: ▪ **Default Tax Reg Number** in Suite Tax account. ▪ **Tax Reg Number** field in Legacy Tax account.

101	cac:TaxScheme				This parent element provides information about the tax scheme used by the customer. This element has no value, but it has child elements that have values associated to it in the template.	This parent element provides information about the tax scheme used by the customer. This element has no value, but it has various child elements that have values associated with it in the template.
102	cbc:ID		Custom Field	Both	This element provides the code of the tax scheme being used. This element has a fixed value VAT.	This element provides the code of the tax scheme being used. This element has a fixed value VAT.
103	cac:PartyLegalEntity				This parent element provides legal information about the buyer. This element has no value, but it has child elements that have values associated to it in the template.	This parent element provides legal information about the buyer. This element has no value, but it has child elements that have values associated with it in the template.
104	cbc:RegistrationName	transaction.billaddressee customer.companyname	Addressee Company Name		This element displays the buyer's full name. If the **Addressee** field has a value in the Billing Address of a transaction, then the full name is displayed along with the element. If not, then the element displays the customer's company name.	This element displays the buyer's full name. If the **Addressee** field has a value in the Billing Address of a transaction, then the full name is displayed along with the element. If not, then the element displays the customer's company name.
105	cbc:CompanyID		Custom Field	Both	This element displays an identifier of the buyer as a legal entity or person. This identifier is issued by an official registrar. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays an identifier of the buyer as a legal entity or person. This identifier is issued by an official registrar. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
106	@schemeID		Custom Value	Both	This element displays the scheme ID of the buyer's legal registration identifier. This element has no value associated with it in the template. You can map the value with the required custom field in the SuiteApp.	This element has the point of contact of the buyer's legal entity or person. This element has no value associated with it in the template. You can map the value with the required custom field in the SuiteApp.
107	cac:Contact				This is parent element has multiple child elements and provides contact information relevant to the buyer. This element has no value associated with it in the template. This element is displayed only if the transaction's customer has a phone number or an email address.	This parent element has multiple child elements and provides contact information relevant to the buyer. This element does not have a value in the template. This element is displayed only if the transaction's customer has a phone number or an email address.
108	cbc:Name		Custom Field	Both	This element has the point of contact of the buyer's legal entity or person. This element has no value associated with it in the template. You can map the value with the required custom field in the SuiteApp.	This element displays the e-mail address of the buyer's contact. This element is displayed in the generated e-document only if the customer's **E-mail Address** field has a value.
109	cbc:Telephone	customer.phone	Phone	Both	This element displays the phone number of the buyer's contact. This element is displayed in the generated e-document only if the customer's **Phone Number** field has a value.	The template does not support this element.
110	cbc:ElectronicMail	customer.email	Email	Both	This element displays the e-mail address of the buyer's contact. This element is displayed in the generated e-document only if the customer's **E-mail Address** field has a value.	The template does not support this element.
111	cac:PayeeParty				This parent element has multiple child elements, with information about the payee. This element is used when the payee is different from the seller. This element has no value in the template.	The template does not support this element.
112	cac:PartyIdentification			Both	This is a parent element and a child of cac:PayeeParty, which has child elements used for identifying the payee party. This element has no value in the template.	The template does not support this element.
113	cbc:ID		Custom Field	Both	This element is used to identify both the payee and the unique banking reference identifier of the payee assigned by the payee's bank. For identifying the payee, you can use the ICD code list. You can use SEPA code for identifying the SEPA bank assigned creditor reference. For example, FR932874294.	The template does not support this element.

					You can map the value with the required custom field in the SuiteApp.	
114	@schemeID		Custom Value	Both	This element used with the cbc:ID element identifies the payee's scheme ID. This has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
115	cac:PartyName				This is a parent element and a child of cac:PayeeParty element. This element provides information about the payee name. This element has no value in the template.	The template does not support this element.
116	cbc:Name		Custom Field	Both	This element displays the name of the payee. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
117	cac:PartyLegalEntity				This parent element has legal information about the payee. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
118	cbc:CompanyID		Custom Field	Both	This element identifies the payee's legal registration. The identifier is issued by an official registrar and identifies the payee as a legal entity or person. For example, FR932874294. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
119	@schemeID		Custom Value	Both	This element is used with cbc:CompanyID and has the scheme ID of the payee's legal registration identifier. For example: 0002 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
120	cac:TaxRepresentativeParty			Both	This parent element has information about the seller's tax representative. This element has no value, but it has child elements that have values associated to it in the template.	The template does not support this element.
121	cac:PartyName				This is a parent element and a child element of cac:TaxRepresentativeParty. This element displays the name of the seller's tax representative. This element has no value, but it has child elements that have values associated to it in the template.	The template does not support this element.
122	cbc:Name		Custom Field	Both	This element displays the full name of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
123	cac:PostalAddress			Both	This is a parent element and a child element of cac:TaxRepresentativeParty. The element has no value, but it has child elements with information about the postal address of the tax representative in template. The required address fields must be filled out to comply with legal requirements.	The template does not support this element.
124	cbc:StreetName		Custom Field	Both	This element displays the main address line of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
125	cbc:Additiona		Custom Field	Both	This element displays the value of an additional address line of the main address	The template does not support this element.

	lStreet Name				of a seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	
126	cbc:CityName		Custom Field	Both	This element displays the common name of the city, town, or village of the tax representative's location. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
127	cbc:PostalZone		Custom Field	Both	This element identifies an addressable group of properties according to the relevant postal service of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
128	cbc:CountrySubentity		Custom Field	Both	This element has information about a country's subdivision of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
129	cac:AddressLine			Both	This element is a parent element and a child of cac:PostalAddress, with information about the additional address line of the seller's tax representative main address. This element has clarifications on the addresses. It has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
130	cbc:Line		Custom Field	Both	This element has information about the additional address line of a main address. It gives information about the seller's tax representative's main address. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
131	cac:Country			Both	This element is a parent element and child of cac:PostalAddress, with information about the country of the address. This element has no value, but it has child elements that have values associated to it in the template.	The template does not support this element.
132	cbc:IdentificationCode		Custom Field	Both	This element displays the ISO code of the seller's tax representative country. This element has no a value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
133	cac:PartyTaxScheme			Both	This is a parent element with information about the VAT party identifier. This element has no value, but it has child elements with values associated with it in the template.	The template does not support this element.
134	cbc:CompanyID		Custom Field	Both	This element represents the VAT identifier of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
135	cac:TaxScheme				This parent element displays the tax scheme used by the tax representative. This element has no value in the template.	The template does not support this element.
136	cbc:ID		Custom Field	Both	This element displays the name of the tax scheme and has a required value VAT, added in the template.	The template does not support this element.
137	cac:Delivery				This parent element has information about the delivery time and location of the invoiced goods and services. This element has no value in the template, but it has child	This parent element has information about the delivery time and location of the invoiced goods and services. This element has no value in the template, but it has various child elements with

					elements with information about the delivery address in the template.	information about the delivery address in the template.
138	cbc:ActualDeliveryDate	transaction.trandate	Date	Both	This element displays the date when the supply of goods and services is completed. This element is displayed along with the value on the generated e-document only if the value is provided in the **Date** field of the transaction. This date must be in YYYY-MM-DD format. You can map the value with the required custom field in the SuiteApp.	This element displays the date when supply of goods and services is completed. This element is displayed along with the value on the generated e-document only if the value is provided in the **Date** field of the transaction. This date must be in YYYY-MM-DD format. You can map the value with the required custom field in the SuiteApp.
139	cac:DeliveryLocation			Both	This element has information about the location where goods and services were delivered. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element has information about the location to which the goods and services are delivered. This element has no value in the template. This element is not printed if the shipping country is not available in the transaction.
140	cbc:ID		Custom Field	Both	This element displays the identifier for the delivery location of the goods and services. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
141	@schemeID		Custom Value	Both	This element is a part of cbc:ID element and displays the scheme ID of the **Deliver to Location** field. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this value.
142	cac:Address			Both	This parent element has information about the address where invoiced goods or services are delivered. This element has no value in the template, but it has child elements that have values associated with it.	This parent element has information about the address to which the invoiced goods or services are delivered. This element has no value in the template, but it has various child elements that have values associated with it.
143	cbc:StreetName	transaction.shipaddr1	Address 1 (Transaction's address line 1)	Both	This element displays the main address line in the transaction's shipping address. This element has the value of **Address 1** field in the transaction's shipping address. This element is not displayed if the **Address 1** field has no value in the transaction.	This element displays the main address line in the transaction's shipping address. This element has the value of **Address 1** field in the transaction's shipping address. The element and its value are not displayed if the **Address 1** field has no value in the transaction.
144	cbc:AdditionalStreetName	transaction.shipaddr2	Address 2 (Transaction's address line 2)	Both	This element displays the main address line in the transaction's shipping address. This element has the value of the **Address 2** field in the transaction's shipping address. The element and its value are not displayed if the **Address 2** field has no value.	This element displays the main address line in the transaction's shipping address. This element has the value of the **Address 2** field in the transaction's shipping address. The element and its value are not displayed if the **Address 2** field has no value.
145	cbc:CityName	transaction.shipcity	City (Transaction's shipping address's city)	Both	This element displays the common name of the city, town, or village of the delivery location in the shipping address. This element has the value of the **City** field in the transaction's shipping address. The element and its value are not displayed if the **City** field has no value.	This element displays the common name of the city, town or village of the delivery location in the shipping address. This element has the value of the **City** field in the transaction's shipping address. The element and its value are not displayed if the **City** field has no value.
146	cbc:PostalZone	transaction.shipzip	Zip (Transaction's shipping address's zip)	Both	This element displays an identifier for a group of properties based on their postal service. This element has the value of the **Zip** field in the transaction's shipping address. The element and its value are not displayed if the **Zip** field has no value.	This element displays an identifier for a group of properties based on their postal service. This element has the value of the **Zip** field in the transaction's shipping address. The element and its value are not displayed if the **Zip** field has no value.
147	cbc:CountrySubentity	transaction.shipstate	State (Transaction's shipping address's state)	Both	This element displays the subdivision of a country in the shipping address. This element has the value of the **State** field in the transaction's shipping address. The element and its value are not displayed if the **State** field has no value.	This element displays the subdivision of a country in the shipping address. This element has the value of the **State** field in the transaction's shipping address. The element and its value are not displayed if the **State** field has no value.
148	cac:AddressLine			Both	This parent element displays additional information about the shipping address. This element has no value but has child elements with values in the template. This parent element is not displayed if its child elements do not have values in the template.	This parent element displays additional information about the shipping address. This element has no value in the template but has a child element with value in **Address 3** field of the transaction's shipping address. This element is not displayed if there is no value in **Address 3**.

149	cbc:Line	transaction.addr3	Address 3 (Transaction's address line 3)	Both	This is a parent element and a child of cac:AddressLine, with information about the additional line of an address, supporting the primary address. This element contains value of **Address 3** field of the transaction's shipping address in the template. This element is not displayed if the **Address 3** field has no value. You can map the value with the required custom field in the SuiteApp.	This element is the child of cac:AddressLine with information about additional address line to support the main line. This element contains value of **Address 3** field of the transaction's shipping address in the template. This element is not displayed if there is no value in **Address 3** field.
150	cac:Country			Both	This parent element displays the country of the shipping address. This element has no value, but it has child elements with values in the template.	This parent element displays the country of the shipping address. This element has no value, but it has child element with value of the **Country** field in the template. This element is not printed if there is no value in the **Country** field.
151	cbc:IdentificationCode	transaction.shipcountry	Country (Transaction 's shipping addresse's country)	Both	This element displays the ISO code of the country in the transaction's shipping address. It displays the value of the **Country** field in the template.	This element displays the ISO code of the country in transaction's shipping address. It displays the value of the **Country** field in the template.
152	cac:DeliveryParty			Both	This parent element displays the details of the delivery party. This element has no value but has a child element with values associated to it in the template.	This parent element displays the details of the delivery party. This element has no value but has a child element with values associated to it in the template.
153	cac:PartyName			Both	This is a parent element and a child of cac:DeliveryParty, with information about the delivery party's name. This element has no value but has a child elements with values associated to it in the template.	This is a parent element and a child of cac:DeliveryParty with information of the delivery party's name. This element has no value but has a child element with values associated to it in the template.
154	cbc:Name	transaction.shipaddressee	Addressee (Transaction 's shipping address's country)	Both	This element displays the name of the customer or party whom the goods and services are to be delivered. This element displays the value of the **Addressee** field in the transaction's shipping address.	This element displays the name of the customer or party to which the goods and services are to be delivered. This element displays the value of the **Addressee** field in a transaction's shipping address.
155	cac:PaymentMeans			Both	This parent element provides a group of business terms by using different child elements to give payment information about an invoice or refunds in a credit memo. This element has no value, but it has various child elements that have values associated with it in the template.	This parent element provides a group of business terms by using different child elements to give information about the payments or refunds. This element has no value, but it has various child elements that have values associated with it in the template.
156	cbc:PaymentMeansCode		Custom Field	Both	This element displays the payment type code for a payment made or to be made. This element has no value in the template. Example: 30 The following list has the details of the Payment Types and it's associated code: ■ Credit Transfer – 30 ■ Direct Debit – 49 ■ 0postgiro – 50 ■ Credit Card – 54 ■ Debit Card – 55 ■ Bankgiro – 56 ■ Sepa Credit Transfer – 58 ■ Sepa Direct Debit – ■ Online Payment Service – 68 ■ Finland – 93, 94, 95. ■ Instrument not defined – 1 ■ National or Regional Clearing – 9 ■ Cash – 10 You can map the value with the required custom field in the SuiteApp.	This element displays the payment type code for a payment made or to be made. This element has no value in the template. Example Value: 30 Example: 30 The following list has the details of the Payment Types and it's associated code: ■ Credit Transfer – 30 ■ Direct Debit – 49 ■ 0postgiro – 50 ■ Credit Card – 54 ■ Debit Card – 55 ■ Bankgiro – 56 ■ Sepa Credit Transfer – 58 ■ Sepa Direct Debit – ■ Online Payment Service – 68 ■ Finland – 93, 94, 95. ■ Instrument not defined – 1 ■ National or Regional Clearing – 9 ■ Cash – 10 You can map the value with the required custom field in the SuiteApp.
157	@name		Custom Value	Both	This element displays the name of the mode of payment. This element has no value in the template.	This element displays the name of the payment method used or to be used to make a payment. This element has no value in the template.

					Example: Credit Transfer You can map the value with the required custom field in the SuiteApp.	Example: Credit Transfer You can map the value with the required custom field in the SuiteApp.
158	cbc:PaymentID		Custom Field	Both	This element displays a text value for associating the payment and the invoice issued by the seller. This is used for creditors' critical reconciliation information. The value helps the seller assign an incoming payment to the relevant payment process. This element has no value in the template. A custom field with a payment reference can be used for this element.	This element displays a text value used to establish a link between the payment and the invoice issued by the seller. It used for creditor's critical reconciliation information. The value helps the seller to assign an incoming payment to the relevant payment process. This element has no value in the template. A custom field with payment reference can be used for this element. Example: 432948234234234 You can map the value with the required custom field in the SuiteApp.
159	cac:CardAccount				This parent element represents a group of business terms with information about the card used to make a payment against an issued invoice. If a buyer made a payment using a card such as a credit or debit card, then the information about the Production account Number (PAN) is mentioned in the invoice. This element has no value in the template, but it has various child elements that have values associated with it. The element is displayed only if there are any payments made using cards or if the refund method in the Credit Memo is a card.	This parent element represents a group of business terms with information about the card used to make a payment against an issued invoice. If a buyer made a payment using a card such as credit or debit card, then the information about the Production account Number (PAN) is mentioned in the invoice. This element has no value, but it has various child elements that have values associated with it in the template. This element is printed only if payments are made using card in Invoice or if the refund method used in Credit Memo is Card.
160	cbc:PrimaryAccountNumberID	payment.cardNumber (Refer CDS)		Both	This element represents the Production account Number (PAN) of the card used for payment. As per card payment security standards, an invoice must not indicate the full PAN of the card. Example: 1234 The last 4 to 6 digits of the PAN (BT-87) are indicated if payment card information (BG-18) is provided in the invoice. This element displays the last 4 digits of any cards used for payment and uses custom data source plug-in to display this information in the template.	This element represents the Production account Number of the card used to make a payment or refund. In accordance with the card payments security standards, an invoice must not have a full PAN of the card. Example: 1234 The last 4 to 6 digits of the payment card PAN number (BT-87) is present if payment card information (BG-18) is provided in the invoice. This element displays the last 4 digits of the card(s) used for payments or refunds. It uses a custom data source plug-in to display this information in the template.
161	cbc:NetworkID	payment.paymentMethodName (Refer CDS)		Both	This element displays the card network identifier such as VISA, American Express, Card. Example: VISA The payment method name is displayed in this element using a custom data source plug-in.	This element displays the card network identifier such as VISA, American Express, Card. Example: VISA The payment method name or the refund method is displayed in this element using custom data source plug-in.
162	cbc:HolderName	payment.cardHolderName (Refer CDS)		Both	This element displays the name of the holder of the card. Example: John Doe This element uses a custom data source plug-in to display the name of the cardholder.	This element displays the cardholder's name. Example: John Doe This element displays the card holder name by using the custom data source plug-in.
163	cac:PayeeFinancialAccount			Both	This parent element displays a group of business terms specifying the credit transfer payments. This element has no value, but it has various child elements with values associated to it in the template.	This parent element displays a group of business terms specifying the credit transfer payments. This element has no value, but it has various child elements that have values associated with it in the template.
164	cbc:ID		Custom Field	Both	This element identifies a payment account. It displays a unique ID of the financial payment account for a payment service provider to which the payment is made, such as IBAN or BBAN. Example: NO99991122222	This element identifies a payment account. It displays a unique ID of the financial payment account for a payment service provider to which the payment is made, such as IBAN or BBAN. Example: NO99991122222 This element has no value in the template.

					This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	You can map the value with the required custom field in the SuiteApp.
165	cbc:Name		Custom Field	Both	This element displays the payment account name of the service provider to which the payment is made. Example: Payment Account This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
166	cac:FinancialInstitutionBranch			Both	This parent element displays the details of a payment service provider. This element has no value, but it has various child elements with values associated to it in the template.	This parent element displays the details of a payment service provider. This element has no value, but it has various child elements that have values associated with it in the template.
167	cbc:ID		Custom Field	Both	This element displays an ID for payment service provider where a payment account is located such as BIC or a national clearing code. Identification scheme identifier is not used in this element. Example: 9999 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays an ID for payment service provider where a payment account is located such as BIC or a national clearing code. Identification scheme identifier is not used in this element. Example: 9999 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
168	cac:PaymentMandate			Both	This parent element displays a group of business terms for direct debit transactions. This element has no value, but it has various child elements with values associated to it in the template.	The template does not support this element.
169	cbc:ID		Custom Field	Both	This element displays a unique identifier assigned by the payee for referencing a direct debit mandate. This is used to notify the buyer of a SEPA direct debit in advance. Example: 123456 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
170	cac:PayerFinancialAccount			Both	This parent element has no value, but it has various child elements with values associated to it in the template. It has information about the payer's financial account.	The template does not support this element.
171	cbc:ID		Custom Field	Both	This element displays an ID with information about the account to be debited by direct debit. Example: 12345676543 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
172	cac:PaymentTerms			Invoice	This parent element has information about the payment terms applied to the due amount. This parent element has no value, but it has a child element with value associated with it in the template.	This parent element has information about the payment terms applied on the due amount. This element has no value, but it has a child element that has values associated with it in the template.
173	cbc:Note		Terms	Invoice	This parent element displays text describing the payment terms that apply to the amount due for payments. It can also describe applicable penalties. If an amount is due for payment (BT-115), then the payment due date (BT-9) or the payment terms (BT-20) is indicated. This element displays a value in the **Terms** field on a transaction. Example: Net within 30 days	This parent element displays a description text of the payment terms that apply on the due amount of the payment. It can also include a description of applicable penalties. If an amount is due for payment (BT-115), then the payment due date (BT-9) or the payment terms (BT-20) is present. This element displays a value in the **Terms** field for a transaction. Example: Net within 30 days

174	cac:AllowanceCharge			Both	This parent element displays a group of business terms with information about applicable allowances of a complete invoice. This element also displays a group of business terms with information about charges and taxes excluding VAT for a complete invoice. This element has no value and is displayed in the generated e-document only if there is a Header Discount or Shipping Cost associated with the transaction.	This parent element displays a group of business terms with information applicable allowances to a complete invoice. This element also displays a group of business terms with information about charges and taxes excluding VAT for a complete invoice. This element has no value and is displayed in the generated e-document only if it has a Header Discount or Shipping Cost associated with the transaction.
175	cbc:ChargeIndicator	True or False	Fixed value	Both	This element displays the charge type used for a transaction. The generated e-document displays the value true when informing about charges and the value false when informing about allowances. Example: ▪ True for shipping and handling. ▪ False for header discounts.	This element displays the charge type used for a transaction. The generated e-document displays he value true when informing about charges and the value false when informing about allowances. Example: ▪ True for shipping and handling. ▪ False for header discounts.
176	cbc:AllowanceChargeReasonCode		Fixed value	Both	This element displays the document level allowance code or the allowance charge reason code. You can use the UNCL5189 code from the code list for allowance of a subset. For charges, you can use the UNCL7161code from the code list. The document level allowance reason code and document level allowance reason display the same reason. Example: 95 for Discount SAA for Shipping and Handling This element does not have a value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays the document level allowance code or the allowance charge reason code. You can use the UNCL5189 code from a code list for allowance of a subset. For charges, you can use the UNCL7161code from the code list. The document level allowance reason code and document level allowance reason display the same reason. Example: 95 for Discount SAA for Shipping and Handling
177	cbc:AllowanceChargeReason		Fixed value	Both	This element displays document level allowance reason or charge reason in text format. The document level allowance reason code and the document level allowance reason, display the same allowance reason. This element displays the following values: ▪ Shipping and handling for charges ▪ Discount for allowances	This element displays document level allowance reason or charge reason in text format. The document level allowance reason code and the document level allowance reason display the same allowance reason. This element displays the following values: ▪ Shipping and handling for charges ▪ Discount for allowances
178	cbc:MultiplierFactorNumeric	transaction.discountrate	Rate (Header discount)	Both	This element displays the document level allowance or charge percentage. It is displayed only for header discounts and uses the value of the **Rate** field. The percentage is used with the document level allowance base amount to calculate the document level or charge amount. the value must be 20. Example: 20 This element has no value associated with Shipping and Handling. It displays the discount rate in the generated e-document retrieved from a transaction.	This element displays the document level allowance or charge percentage. This element is displayed only for header discounts and uses the value of the **Rate** field in a transaction. The percentage is used with the document level allowance base amount to calculate the document level or charge amount. For 20 percent the value must be 20. Example: 20
179	cbc:Amount	transaction.shippingcost+ transaction.handlingcost (For shipping and handling) transaction.discounttotal (for discount)	Discount Item	Both	This element displays the document level allowance or charge amount without VAT. This value must be rounded off to a maximum of 2 decimal. This element uses the following values from the Summary section of a transaction in the template: ▪ The sum of shipping and handling costs for charges	This element displays the document level allowance or charge amount without VAT. The value must be rounded maximum to 2 decimals. This element uses the following values from the Summary section of a transaction in the template: ▪ The sum of shipping and handling costs for charges ▪ The value of Discount Item field for allowances.

					▪ The value of Discount Item field for allowances. Example: 200	Example: 200
180	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with the cbc:Amount element, to display the ISO code of a transaction's currency.	This element is used with the cbc:Amount element, to display the ISO code of a transaction's currency.
181	cbc:BaseAmount	transaction.subtotal	Subtotal	Both	This element displays the base amount along with the document level allowance or charge percentage. It calculates the document level allowance or charge amount. The value is rounded off to 2 decimal places in the template. A value is provided in the **Subtotal** field in the Summary section of a transaction Example value: 200	This element displays the base amount along with the document level allowance or charge percentage. It calculates the document level allowance or charge amount. The value is rounded off to 2 decimal places in the template. A value is provided in the **Subtotal** field present in the summary section of a transaction. This element is displayed only if the header discount is available. Example value: 200
182	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:BaseAmount element, to display the currency's ISO code used by a transaction.	This element is used with cbc:BaseAmount element to display the currency's ISO code used by a transaction.
183	cac:TaxCategory			Both	This parent element displays the unique tax category and its details. This element has no value, but it has child elements with values associated to it in the template.	This parent element displays the unique tax categories and their details. This element has no value, but it has a child element that has values associated with it in the template.
184	cbc:ID		Custom Field	Both	This element displays a code to identify the VAT category applied to document level allowance or charge. Example: Value: S In EMEA, the following codes are used: ▪ AE - Vat Reverse Charge – Specifies the standard VAT rate reversed from the invoice. ▪ E - Exempt from Tax – Specifies that taxes are not applicable ▪ S – Standard Rate – Specifies the standard rate ▪ Z – Zero Rated Goods – Specifies that the goods are at zero rate ▪ G - Free Export Item, VAT Not Charged – Specifies that the item is free export and taxes are not charged ▪ K – VAT exempt for EEA intra-community supply of goods and services – A tax category code that identifies of the item is VAT exempted due to intra community supply in the European Economic Area This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays a code to identify the VAT category applied to document level allowance or charge. Example: Value: S In EMEA, the following codes are used: ▪ AE - Vat Reverse Charge – Specifies the standard VAT rate reversed from the invoice. ▪ E - Exempt from Tax – Specifies that taxes are not applicable ▪ S – Standard Rate – Specifies the standard rate ▪ Z – Zero Rated Goods – Specifies that the goods are at zero rate ▪ G - Free Export Item, VAT Not Charged – Specifies that the item is free export and taxes are not charged ▪ K – VAT exempt for EEA intra-community supply of goods and services – A tax category code that identifies of the item is VAT exempted due to intra community supply in the European Economic Area This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
185	cbc:Percent	For LegacyTax account: ▪ transaction.shippingtax1rate ▪ taxCodeDet.taxRate For SuiteTax account: ▪ custom.shippingTaxRatesOfSuiteTax ▪ suiteTaxCodeDet.taxRate	Shipping Tax Rate	Both	This element displays the document level allowance or VAT charge rate. The VAT rate is displayed as the percentage that applies to the document level allowance or charge. The categories include the following: ▪ Tax rates of the unique tax codes when displaying header discounts. ▪ Tax rates of shipping when displaying shipping and handling charges. It uses Custom Data Source plug-in to display values. Example: 25 This element displays the value in the **Shipping Tax Rate** field. This element is not displayed if the field has no value. You can map the value with the required custom field in the SuiteApp.	This element displays the document level allowance or VAT charge rate. The VAT rate is displayed as the percentage that applies to the document level allowance or charge. The categories include the following: ▪ Tax rates of the unique tax codes when displaying header discounts. ▪ Tax rates of shipping when displaying shipping and handling charges. It uses custom data source plug-in to display the values. Example: 25

186	cac:TaxScheme			Both	This parent element displays the details of the tax scheme used in a transaction. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This parent element displays the details of the tax scheme used in a transaction. This element has no value in the template.
187	cbc:ID		Fixed Value	Both	This element displays the name of the tax scheme used in a transaction. A fixed VAT value is provided in the template for this element.	This element displays the name of the tax scheme used in a transaction. A fixed value VAT is provided in the template for this element.
188	cac: Tax Total			Both	This parent element displays the total tax details including the amounts for a transaction. When a tax currency code is provided, two instances of tax total must be present and only one instance with tax subtotal must be used. This element does not have a value in the template. This element has no value in the template.	This parent element displays the total tax details including the amounts for a transaction. When a tax currency code is provided, two instances of tax total must be present and only one instance with tax subtotal must be used. This element has no value in the template.
189	cbc:TaxAmount	transaction.taxtotal	Tax	Both	This element displays the total VAT amount for an invoice or the total VAT amount in the accounting currency as required by the seller. This value is rounded off to a maximum of 2 decimal places in the template. This element prints the value of the **Tax Total** field from the Summary section of the invoice.	This element displays the total VAT amount for an invoice or the total VAT amount present in the accounting currency as required by the seller. This value is rounded off to maximum 2 decimals in template. This element prints the value of the **Tax Total** field from the Summary section of the invoice.
190	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxAmount element, to display the ISO code of the currency used by the transaction.	This element is used with cbc:TaxAmount element to display the ISO code of the currency used by the transaction.
191	cac:TaxSubtotal			Both	This parent element displays a group of business terms with information about VAT break down based on different categories, rates and exemption reasons. This element has no value, but it has child elements with values associated with it.	This parent element displays a group of business terms with information about VAT break down on basis of different categories, rates, and exemption reasons. This element has no value, but it has various child elements with values associated to it in the template.
192	cbc:TaxableAmount	taxDetails.taxableAmount (refer CDS)	Amount of all items calculated per unique tax category	Both	This element displays all the taxable amount for each item specific to a unique VAT category code. The element uses custom data source plug-in to get the values. It also adds the shipping cost to the taxable amount as required. This value must be rounded off to 2 decimal places.	This element displays all the taxable amounts for each item specific to a unique VAT category code or tax code. It uses custom data source plug-in to get the values. The element also adds the shipping cost wherever required in the taxable amount. This value must be rounded off to 2 decimals.
193	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxableAmount element to display the ISO code of the transaction's currency.	This element is used with cbc:TaxableAmount element to display the ISO code of the transaction's currency.
194	cbc:TaxAmount	taxAmount ((Using CDS)	Tax amount of all items calculated per unique tax category	Both	This element displays the total VAT amount for each item specific to a unique VAT category code. The element uses custom data source plug-in to get the values. The sum of all VAT amounts of the Unique VAT category is equal to **Total Tax** field in the Summary section. The value is rounded off to a maximum of 2 decimal places in the template.	This element displays the total VAT or Tax amount for each item specific to a unique VAT category code or tax code. It uses custom data source plug-in to get the values. The value is rounded off to a maximum of two decimals in the template.
195	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxAmount to display the ISO code of the transaction's currency.	This element is used with cbc:TaxAmount to display the ISO code of the transaction's currency.
196	cac:TaxCategory			Both	This parent element displays the VAT category details. This element has no value, but it has child elements with values associated with it.	This parent element displays the VAT category details. This element has no value, but it has various child elements that have values associated with it in the template
197	cbc:ID		Custom Field	Both	This element displays a code to identify the VAT category. Example: Value: S	This element displays a code to identify the VAT category. Example: Value: S

					In EMEA, the following codes are used: ■ AE - Vat Reverse Charge – Specifies the standard VAT rate reversed from th ■ E - Exempt From Tax – Specifies that taxes are not applicable ■ S – Standard Rate – Specifies the standard rate. ■ Z – Zero Rated Goods – Specifies that the goods are at zero rate ■ G - Free Export Item, VAT Not Charged – Specifies that the item is free to export and taxes are not charged. ■ K – VAT Exempt for EEA intra-community Supply of Goods and Services – A tax category code that identifies of the item is VAT exempted due to intra community supply in the European Economic Area. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	In EMEA, the following codes are used: ■ AE - Vat Reverse Charge – Specifies the standard VAT rate reversed from th ■ E - Exempt From Tax – Specifies that taxes are not applicable ■ S – Standard Rate – Specifies the standard rate. ■ Z – Zero Rated Goods – Specifies that the goods are at zero rate ■ G - Free Export Item, VAT Not Charged – Specifies that the item is free to export and taxes are not charged. ■ K – VAT Exempt for EEA intra-community Supply of Goods and Services – A tax category code that identifies of the item is VAT exempted due to intra community supply in the European Economic Area. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
198	cbc:Percent	In LegacyTax account: ■ taxDetails.taxRate ■ custom.handlingTaxRate ■ custom.shippingTaxRate In SuiteTax account: ■ taxDetails.taxRates ■ custom.handlingTaxRate ■ custom.shippingTaxRate	Tax Rate	Both	This element displays the VAT category rate for each unique tax code used by items for shipping. This element is not displayed if a VAT category has no rate. The VAT rate is displayed as percentage for the applicable VAT category. This uses custom data source plug-in to retrieve the values. In SuiteTax accounts, if the tax type has multiple tax rates, then rates are also displayed for single tax type. Example: 25	This element displays the VAT category rate for each unique tax code used in items and shipping. The VAT rate is displayed as a percentage that applies to the applicable VAT category. It uses custom data source plug-in to get the values. In SuiteTax accounts, if a tax type has multiple tax rates, then multiple tax rates are displayed for a single tax type. Example: 25
199	cbc:TaxExemptionReasonCode		Custom Field	Both	This element displays a VAT exemption reason code. A coded statement indicates the reason for exempting the VAT amount. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays a VAT exemption reason code. A coded statement is provided explaining the reason for exempting the VAT amount. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
200	cbc:TaxExemptionReason		Custom Field	Both	This element displays the VAT exemption reason text. Example: Exempt This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays the VAT exemption reason text. A statement in text is provided explaining the reason for exempting the VAT amount. Example: Exempt This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
201	cac:TaxScheme				This parent element displays the tax scheme. This element has no value, but it has child elements with values associated with it in the template.	This parent element displays tax scheme. This element has no value, but it has various child elements that have values associated with it in the template.
202	cbc:ID		Fixed Value	Both	This element displays the name of the tax scheme used in a transaction. A fixed value VAT is assigned to this element.	This element displays the name of the tax scheme used in a transaction. A fixed value VAT is assigned in a template.
203	cac:LegalMonetaryTotal			Both	This parent element displays the document totals. This element has no values, but it has child elements that provide monetary totals for an invoice or a credit memo in a template.	This parent element displays the document totals. This element has no value, but it has various child elements that have values associated with it in the template.

204	cbc:LineExtensionAmount	transaction.subtotal	Subtotal	Both	This element displays the sum of all the invoice line net amounts in an invoice. The value is rounded off to a maximum of 2 decimal places in the template. This element displays the **Subtotal** field value from the transaction's Summary tab.	This element displays the sum of all the invoice line net amounts in an invoice. This value is rounded off to maximum 2 decimals in template. This element displays the **Subtotal** field value from the transaction's Summary tab.
205	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:LineExtensionAmount element to display the ISO code of a transaction's currency.	This element is used with cbc:LineExtensionAmount element to display the ISO code of a transaction's currency.
206	cbc:TaxExclusiveAmount	transaction.altshippingcost + transaction.althandlingcost + transaction.subtotal – transaction.discounttotal	Subtotal + Allowances – Discount	Both	This element displays the total amount of an invoice without VAT. The value of this element must be rounded off to two decimals. In a template, you must add the transaction's amount in the Subtotal field in the Summary tab with the handling and shipping charges to get a value. Example: 3600.0 You can add extra allowances and change the template to accommodate this element.	This element displays the total amount of an invoice without VAT. The value of this element must be rounded off to two decimals. In a template, you must add the transaction's amount in the Subtotal field in the Summary tab with the handling and shipping charges and subtract the total discount to get a value. Example: 3600.0 You can add extra allowances and change the template to accommodate this element.
207	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxExclusiveAmount to display the ISO code of the transaction's currency.	This element is used with cbc:TaxExclusiveAmount to display the ISO code of the transaction's currency.
208	cbc:TaxInclusiveAmount	transaction.total	Total	Both	This element displays the total amount of a transaction with VAT. The value is displayed in the **Total** field on a transaction's Summary tab. This value is rounded off to 2 decimal places in the template.	This element displays the total amount of transactions with VAT. The value is displayed in the **Total** field on a transaction's Summary tab. This value is rounded off to two decimals in template.
209	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxInclusiveAmount element to display the ISO code of the transaction's currency	This element is used with cbc:TaxInclusiveAmount element to display the ISO code of the transaction's currency.
210	cbc:AllowanceTotalAmount	transaction.discounttotal	Discount Item	Both	This element displays the sum of allowances at the document level. The value of this element is rounded off to 2 decimal places. Example: 200.0	This element displays the sum of allowances at the document level. The value of this element is rounded off to two decimals. Example: 200.00
211	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:AllowanceTotalAmount element to display the ISO code for the transaction's currency.	This element is used with cbc:AllowanceTotalAmount element to display the ISO code for the transaction's currency.
212	cbc:ChargeTotalAmount	transaction.altshippingcost	Shipping Cost	Both	This element displays the shipping cost charge at the document level in the transaction. The value is rounded off to 2 decimal places in the template. Example: 10.0	This element displays the sum of charges at document level in the transaction. The value is rounded to two decimals in the template. It displays the total cost of shipping. Example: 123.23
213	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	The element is used with cbc:ChargeTotalAmount to display the ISO code of the transaction's currency.	The element is used with cbc:ChargeTotalAmount to display the ISO code of the transaction's currency.
214	cbc:PrepaidAmount	transaction.amountpaid	Payments	Both	This element displays the sum of the amount paid in advance, for a transaction. The value is rounded off to a maximum 2 decimal places in the template. This value displays the sum of all the payments made for a transaction.	This element displays the sum of the amount paid in advance for a transaction. The value is rounded off by maximum of two decimals in the template. This value displays the sum of all the payments made for a transaction. Example: 1000.0
215	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PrepaidAmount element to display the ISO code of the transaction's currency.	This element is used with cbc:PrepaidAmount element to display the ISO code of the transaction's currency.
216	cbc:PayableRoundingAmount		Custom Field	Both	This element displays the rounding amount. The amount is added to the invoice total to round off the amount to be paid. The value is rounded off by maximum two decimals in template. Example: 0.0 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.

217	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PayableRoundingAmount element to display the ISO code of the transaction's currency.	This element is used with cbc:PayableRoundingAmount element to display the ISO code of the transaction's currency.
218	cbc:PayableAmount	For Invoice: transaction.amountremainingtotalbox For Credit Memo: transaction.amountremaining	Amount Due	Both	This element displays the amount due for payment and the outstanding amount that is requested to be paid. The value is rounded off to a maximum 2 decimal places in the template. The **Amount Due** field value in the Summary tab is used for the transaction. Example: 3500.0	This element displays the amount due for payment and the outstanding amount requested to be paid. The value is rounded off by maximum two decimals in the template. The **Amount Due** field value in the Summary tab is used for the transaction. Example: 3500.12
219	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PayableAmount element to display the ISO code of the transaction's currency.	This element is used with cbc:PayableAmount element to display the ISO code of the transaction's currency.
220	cac:InvoiceLine			Both	This parent element displays the line item components. It also has a group of business terms providing information about individual transaction lines. This element has no value, but it has child elements with values in the template.	This parent element displays the line item components on Invoice. It also has a group of business terms providing information about individual Invoice lines. This element does not have value, but it has child elements with values in template. This element is used only in Invoice transaction type.
221	cac:CreditNoteLine			Credit Memo	This parent element displays the line item components on a Credit Memo. It also has a group of business terms providing information about individual invoice lines for the credit memo. This element has no value, but it has child elements with values associated to it in the template. This element is used only for the Credit Memo transaction type.	This parent element displays the line item components for the Credit Memo transaction type. It also has a group of business terms providing information about individual Invoice lines for Credit Memo transactions. This element has no value, but it has various child elements that have values associated with it in the template. This element is used only in Credit Memo transaction type.
222	cbc:ID	item.line	Line Sequence Number	Both	This element displays the transaction line ID for each item. It has a unique identifier for an individual line in the transaction. The line sequence number of a line item is used in the template. Example: 12	This element displays the transaction line ID for each item. It has a unique identifier for individual line within the transaction. The line sequence number of a line item is used in the template. Example: 12
223	cbc:Note	transaction.memo	Memo	Both	This element is used in the invoice line note. It is a text note with unstructured information relevant to the invoice line. The header level **Memo** field is used as value in the template. Example: New article number 12345	This element is used in invoice line note. It is a text note giving unstructured information relevant to the invoice line. The header level **Memo** field is used as value in the template. Example: New article number 12345
224	cbc:InvoicedQuantity	item.quantity	Quantity	Both	This element displays the value of invoiced quantity of each item in the transaction line. The **Quantity** field on the line item is used as value in the template. Example: 100 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.	This element displays the value of invoiced quantity of each item in the invoice line. The value of the **Quantity** field in the line item is used in the template. This element is not printed if the **Quantity** field has no value in the item. This element is used only in Invoice transaction type. Example: 100 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.
225	cbc:CreditedQuantity	item.quantity	Quantity	Credit Memo	This element displays the value of invoiced quantity of each item in the credit note line. The **Quantity** field on the line item is used as value in the template. If the **Quantity** field for the item has no value, this element will not be printed. This element is used only for the Credit Memo transaction type. Example: 100 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.	This element displays the value of the credited quantity of each item in the credit note line. The value of the **Quantity** field in the line item is used in the template. This element is not printed if the Quantity field has no value in the item. This element is used only in the Credit Memo transaction type. Example: 100 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.
226	@unitCode		Fixed Value	Both	This element is used with cbc:InvoicedQuantity element, and displays the invoiced quantity for unit of measure. This unit of measure applies to the invoiced	This element is used with cbc:InvoicedQuantity or cbc:CreditedQuantity element and displays the invoiced or credited quantity for the unit of measure. This unit of measure applies to

					quantity. An alphanumeric code consisting of 2 or 3 letters can be used. Example: C62 This element has a fixed value "NAR" in the template. You can map the value with the required custom field in the SuiteApp.	the invoiced quantity. An alphanumeric code consisting of 2 or 3 letters can be used here. Example: C62 This element has a fixed value NAR in the template. You can map the value with the required custom field in the SuiteApp.
227	cbc:LineExtensionAmount	item.amount	Amount	Both	This element displays the total amount of the invoice line. The total amount includes the line level allowances, charges, and other relevant taxes but excludes VAT. The value is rounded off to a maximum of 2 decimal places. The amount of each element is used as the value in the template. This element is not printed if the **Amount** field has no value for an item. Example: 2145.00	This element displays the total amount of the invoice line. The total amount excludes VAT but includes the line level allowances, charges, and other relevant taxes. The value is rounded off to maximum 2 decimals. The amount of each item is used as a value in the **Amount** field in the template. This element is not displayed of the Amount field has no value in an item. Example: 2145.00
228	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:LineExtensionAmount element to display the ISO code of the transaction's currency.	This element is used with cbc:LineExtensionAmount element to display the ISO code of the transaction's currency.
229	cbc:AccountingCost		Custom Field	Both	This element displays the accounting reference of the invoice line buyer. A text value specifies the location of where to book relevant data in the buyer's financial accounts. Example: 1287:65464 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays the accounting reference of the invoice line buyer. A text value specifies the location on where to book relevant data in the buyer's financial accounts. Example: 1287:65464 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
230	cac:InvoicePeriod			Invoice	This is parent element displays a group of business terms providing information about the invoice line period. This element has no value in the template.	The template does not support this element.
231	cbc: Start Date		Custom Field	Invoice	This element displays the invoice line period start date. The format of the invoice line start date is YYYY-MM-DD. Example: 2017-10-05 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
232	cbc: End Date		Custom Field	Invoice	This element displays the invoice line period end date. The format of the invoice line end date is YYYY-MM-DD. Example: 2017-10-15 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
233	cac:OrderLineReference			Both	This parent element displays the reference of the order line. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
234	cbc: LineID		Custom Field	Both	This element displays an object ID of a seller's invoice line. Example: AB12345 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
235	cac:DocumentReference			Both	This parent element displays the details of the line object identifier. This element has no value, but it has child elements with values associated to it in the template.	The template does not support this element.
236	cbc:ID		Custom Field	Both	This element displays the seller's invoice line object identifier.	The template does not support this element.

					Example: AB12345 This element does not have a value in the template. You can map the value with the required custom field in the SuiteApp.	
237	@schemeID		Custom Value	Both	This element is used with cbc:LineID element to display the ID of the scheme identifier of an invoice line object. Example: ABZ This element does not have a value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
238	cbc:DocumentTypeCode		Fixed value 130 for Invoice Fixed Value 130 if Credit Memo is created from Invoice	Both	This element displays the document type code. Code "130" indicates the reference for an invoice object. This code is not used for other documents. In the Invoice template, the code value is always 130. If a credit memo is created from an invoice, the code is always 130 in the credit memo template. Default Value: 130 You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
239	cac:AllowanceCharge			Both	This parent element displays a group of business terms with information about allowances or charges of an individual invoice. This element has no value, but it has child elements with values associated with it in the template.	The template does not support this element.
240	cbc:ChargeIndicator		If line item is of type Discount, display "false" For other type of items, display "true"	Both	This element displays "true" or "false" value for the item type in a transaction. If the item type is discount, the value is "false". For item types other than discount, the value is "true" in the generated e-document. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
241	cbc:AllowanceChargeReasonCode		If line item is of type Discount, the value 95 is displayed For other type of items, display no value	Both	This element displays the line level allowance or charge reason code. If the line item type is Discount, the value of the code is 95; otherwise, the element will not have a value in the template. Example: 95 You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
242	cbc:AllowanceChargeReason		If item type is Discount then display "Discount" For other type of items, the element has no value	Both	This element displays the line level allowance or charge reason in text. If the line item type is Discount, the value of the field is **Discount** in template; otherwise, the element is not displayed. Example: Discount You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
243	cbc:MultiplierFactorNumeric	item.rate	If item type is Discount, the values of Unit Price or Rate are displayed. For other type of items, the	Both	This element displays the line level allowance or charge percentage depending on the line level allowance base amount. It calculates the line level allowance or charge amount only for discount item type. For discount item type, the unit price or rate value of the item is displayed. If item type is not discount, the element is not available in the generated e-document.	The template does not support this element.

			element has no value. Unit Price (Item) or Rate (Item).		For discount item type, the unit price or rate value of the item is displayed. If unit price or rate is in percentage, the value will be in units. Example: 20% will be 20 If unit price or rate is not percentage, the value will be in decimals. Example: 20 will be 20.00 You can map the value with the required custom field in the SuiteApp.	
244	cbc:Amount	item.amount	If item type is Discount the Amount value is dis played. For other item types, the element has no value. Amount	Both	This element displays the line level allowance or charge amount without VAT. The value of this element is used from the value of **Amount** field on a transaction's item. The value is rounded off to maximum two decimals. For Discount item type, the value is displayed in **Amount** field of a line item. For other item types, this element has no value. Example: 200 You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
245	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:Amount element to display the ISO code of the transaction's currency.	The template does not support this element.
246	cbc:BaseAmount		Custom Field	Both	This element displays the line level allowance or charge base amount. The base amount is used according to the line level allowance or charge percentage. This is to calculate the line level allowance or charge amount. The value is rounded off to maximum two decimals. Example: 1000	The template does not support this element.
247	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:BaseAmount element to display the ISO code of the transaction's currency.	The template does not support this element.
248	cac:Item			Both	This parent element displays a group of business terms with information about invoiced goods and services for all the item types except Description type items. This element has no value, but it has child elements with values associated with it in the template.	This parent element displays a group of business terms with information about invoiced goods and services. The element has no value but has various child elements that have values associated with it in the template. This element is not displayed for the Description item type.
249	cbc:Description	item.description	Description	Both	This element displays the description and features of each item in a transaction. The value of the **Description** field for items is displayed. Example: Long description of an item	This element displays the description and features for each item in a transaction. The value of the **Description** field of items is displayed. Example: Long description of an item.
250	cbc:Name	item.name	Item Name	Both	This element displays the item name entered in the **Item Name** field in a transaction. Example: Item name	This element displays the item name. Example: Item name
251	cac:BuyersItemIdentification			Both	This parent element displays the buyer's item identification and has no value in the template.	The template does not support this element.
252	cbc:ID		Custom Field	Both	This element displays an identifier assigned by the buyer for an item. Example: 123455 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
253	cac:SellersItemIde			Both	This parent element displays the information about seller's identification. This element has no value, but it has child elements with values associated with it in the template.	This parent element displays the information about seller's identification. This element has no value in the template.

	ntification					
254	cbc:ID	item.item	Item Name or Item Number	Both	This element displays an identifier assigned by the seller for an item. The element displays the values of the **Item Name** or **Item Number** fields or both. The value is displayed only for items without discount and description in the generated e-document. Example: 9873242	This element displays an identifier assigned by the seller for an item that has the value of the **Item Name** or **Number** or both the fields. This is only for items that do not have discount and description in the generated e-document. Example: 9873242
255	cac:StandardItemIdentification			Both	This parent element has standard item identification details. This element has no value, but it has child elements with values associated with it in the template. This Parent element is displayed only if the item has UpcCode.	This element has the standard item identification details. The element has no value but has various child elements that have values associated with it in the template. This element is not displayed If the item does not have a UPC code.
256	cbc:ID	item1.itemUpc Code (Refer CDS)	UPC Code	Both	This element displays the standard item identifier based on a registered scheme. It displays the value of the **UPC Code** field on the Item Record Example: 10986700 This element uses custom data source plug-in to source the value of an item's UPC code in a template.	This element displays a standard item identifier based on a registered scheme. Example: 10986700 This element uses a custom data source plug-in to source the value of an item's **UPC code** field in a template.
257	@schemeID		Fixed value	Both	This element is used with cbc:ID element, and displays the scheme ID for the item standard identifier. This element has a default value 0160. Example: 0160 You can map the value with the required custom field in the SuiteApp.	This element is used with cbc:ID element and displays the scheme ID for the item standard identifier. Example: 0160 This element has a fixed value 0160 in the template. You can map the value with the required custom field in the SuiteApp.
258	cac:OriginCountry			Both	This parent element is used to display an item's origin country. This element has no value, but it has child elements with values associated with it in the template. This parent element is displayed only if the item has an origin country.	This is parent country is used to display the item's origin country. This is a parent element that has no value. It has child elements with values.
259	cbc:IdentificationCode	item1.itemCountry	Manufacturer Country	Both	This element displays the ISO code of an item's country of origin. This element displays the value of the **Manufacturer Country** field in the Item record. This element is not displayed if the field has no value. Example: CN This element uses custom data source plug-in implementation to source the value of an item's manufacturer country.	This element displays the ISO code of an item's origin country. It is not printed if the item has no value in the **Manufacturer Country** field. Example: CN This element uses custom data source plug-in to source the value of an item's manufacturer country.
260	cac:CommodityClassification			Both	This parent element displays the commodity classification code. This is required when members from same country buy from each other, and the traders need statistical information in the invoice. It is that you use the Item Classification Identifier (BT-158) with the code HS, as an identifier list for this purpose. This element has no value, but has child elements with values associated with it in the template. This element is not printed if there is no value for UPC Code and Manufacturing country in the item.	This parent element displays commodity classification code when members from the same country buy commodities from each other, and the traders need statistical information in the invoice. NetSuite you use the Item Classification Identifier (BT-158) with a code HS as an identifier list for this purpose. The element has no value but has various child elements that have values associated with it in the template. This element is not printed if the **UPC Code** and **Manufacturing Country** fields have no values in the item.
261	cbc:ItemClassificationCode	item1.itemUpc Code	UPC Code	Both	This element displays an item classification identifier code to classify items based on its type or nature. Example: 9873242	This element displays an item classification identifier code to classify items based on its type or nature. Example: 9873242

					This element uses custom data source plug-in to source the value of an item's UPC code.	This element uses custom data source plug-in to source the value of an item's UPC code.
262	@listID		Fixed Value	Both	SME: This element is used with cbc:ItemClassificationCode element and displays a scheme ID for item classification identifier. It has a fixed value HS. Example: STI You can map the value with the required custom value in the SuiteApp	This element is used with cbc:ItemClassificationCode element and displays a scheme ID for item classification identifier. It has a fixed value HS in the template. Example: STI You can map the value with the required custom value in the SuiteApp.
263	@listVersionID		Custom Value	Both	This element is used with cbc:ItemClassificationCode element and displays scheme ID for the item classification identifier. This element has no value in the template.	This element is used with cbc:ItemClassificationCode element and displays scheme ID for the item classification identifier. This element has no value in the template. You can map the value with the required custom value in the SuiteApp.
264	cac:ClassifiedTaxCategory			Both	This parent element displays a group of business terms with information about the VAT applicable for invoiced goods and services in an invoice line. This element has no value, but it has child elements with values associated with it in the template. SuiteTax accounts can have multiple tax rates to display this element and its child element multiple times, except for the Discount item type.	This parent element displays a group of business terms with information about the VAT applicable for invoiced goods and services in an invoice. This element has no value, but it has various child elements that have values associated with it in the template. The SuiteTax accounts can have multiple tax rates that display this element and its child element multiple times, except for the Discount item type.
265	cbc:ID		Custom Field	Both	This element displays the VAT category code for an invoiced item. Example: S This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	This element displays the VAT category code for an invoiced item. Example: S This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
266	cbc:Percent	For Legacy Tax Account: item.taxrate1 For SuiteTax Account: ■ custom.uniqueItems ■ WithMultipleTaxRatesInSTArr (Refer CDS	Tax Rate	Both	This element displays the VAT rate percentage applied to the invoiced item. It displays the tax rate value of an item. This element is not displayed if the **Tax Rate** field in an invoice has no value. Example: 25	This element displays the VAT rate in the percentage applied to the invoiced item. It displays the value of tax rate of the item. Example: 25
267	cac:TaxScheme			Both	This element displays information about tax scheme. This parent element has no value, but it has child elements with values associated with it in the template.	This element displays information about tax scheme. This element has no value, but it has various child elements that have values associated with it in the template.
268	cbc:ID		Fixed Value	Both	This element displays an ID of the tax scheme used in an invoice. A fixed value VAT is added in the template.	This element displays an ID of the tax scheme used in an invoice. A fixed value VAT is added in the template.
269	cac:AdditionalItemProperty			Both	This parent element displays a group of business terms with information about item attributes of invoiced goods and services. This element has no value, but it has various child elements that have values associated to it in the template.	The template does not support this element.
270	cbc:Name		Custom Field	Both	This element displays the name of the item attribute or the property of the item. Example: Color This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.

271	cbc:Value		Custom Field	Both	This element displays a value for the item attribute. Example: Black This element has no value in the template. You can map the value with the required custom field in the SuiteApp.	The template does not support this element.
272	cac:Price			Both	This parent element displays the price details. It has child elements with information about the price applied for the invoiced goods and services in an invoice. This element has no value, but it has various child elements with values in the template.	This parent element displays the price details and has child elements with information about the price applied for the invoiced goods and services in the invoice. This element has no value, but it has various child elements that have values associated with it in the template.
273	cbc:PriceAmount	Item.rate	Unit Price or rate	Both	This element displays the price of an item excluding VAT and item price discount. The item net price must be equal to the item gross price, which is less than the item price discount if both the prices are provided. The item price cannot be negative. The unit price or rate of an item is used as a value and must be rounded off to 2 decimal places. Example: 10.8955 is rounded to 11	This element displays the price of an item excluding VAT and item price discount. The item net price must be equal to the item gross price, which is less than the item price discount if both prices are provided. The item price cannot be negative. The unit price or rate of an item is used as a value and must be rounded to two decimal points. Example: 10.8955 is rounded to 11
274	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PriceAmount to display the ISO code of the transaction's currency.	This element is used with cbc:PriceAmount to display the ISO code of the transaction's currency.
275	cbc:BaseQuantity	item.quantity	Quantity	Both	This element displays quantity of item units to which the price applies. The value is displayed in the item's **Quantity** field. Example: 1 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.	The template does not support this element.
276	@unitCode		Custom Value	Both	This element displays the unit of measure code applicable to the item price base quantity. Also, the unit of measure must be same as the unit code of the invoiced or credited quantity. The unit of measure value has a 3-digit alphanumeric code. This element has no value in the template.	The template does not support this element.
277	cac:AllowanceCharge			Both	This element displays allowance details of an invoice. This element has no value, but it has child elements with values in the template.	The template does not support this element.
278	cbc:ChargeIndicator		Fixed Value		This element indicates the charge. A charge on price level is not valid. This element has a fixed value false.	The template does not support this element.
279	cbc:Amount	item.amount	Amount		This element is used to calculate the item price discount. The total discount is subtracted from the gross item price to calculate the net item price. This element displays the value if only if the item type is discount.	The template does not support this element.
280	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:Amount element to display the ISO code of the transaction's currency.	The template does not support this element.
281	cbc:BaseAmount	item.amount	Item Amount	Both	This parent element displays the gross item price. Item amount is used as value for this element. It is the unit price excluding VAT before subtracting the item price discount. The item price discount value must not be negative. The formula for calculating gross item price is, quantity multiplied by unit, and price divided by rate per Item. Alternatively, the item amount can also be used.	The template does not support this element.

					Example: 123.5	
282	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:BaseAmount element to display the ISO code of the transaction's currency.	The template does not support this element.

ANZ PEPPOL Template: PEPPOL Template for Australia and New Zealand

The following table gives an overview of all the elements which are being used in the generic PEPPOL templates for transaction types- invoice, credit memo and vendor bill. These templates are specifically designed for Australia and New Zealand Localization. Some of the elements below are using hard coded values, some are using the values from the transaction and its related records like subsidiary, customer, and some are being derived from Custom Data Source Plug-in.

Users are suggested to use these templates along with the Custom Data Source plug-in to generate the e-document template for invoice and credit memo or converting inbound e-documents to vendor bill correctly. Also, they can add their own customization in the template to get the generated e-document template in their desired form.

The names and the location of the templates, Custom Data Source in Electronic Invoicing SuiteApp is below:

1. ANZ Generic Invoice Template

   Template Name: ANZ_Invoice_Peppol_Template_Generic.txt

   Template Location: Bundle 436209/Sample Templates
2. ANZ Generic Credit Memo Template

   Template Name: ANZ_Creditmemo_Peppol_Template_Generic.txt

   Template Location: Bundle 436209/Sample Templates
3. ANZ Generic Vendor Bill Template

   Template Name: ANZ_VendorBill_Inbound_Template_Generic.txt

   Template Location: Bundle 436209/Sample Templates
4. Custom Data Source Plug-in (Invoice and Credit Memo)

   File Name: pl_custom_data_source_anz_peppol.js

   Location: Bundle 436209/src/comp/pl

   Plug-in Implementation name: ANZ PEPPOL Custom Data Source Plug-in
5. Custom Data Source Plugin (Vendor Bill)

   File Name: pl_custom_data_source_anz_inbound_peppol.js

   Location: Bundle 436209/src/comp/pl

   Plug-in Implementation name: ANZ PEPPOL Inbound CDS

Details of the Invoice and Credit Memo Outbound Templates

Note: To use the above templates, paste the content into "TEMPLATE FOR OUTBOUND E-DOCUMENTS" field of the new E-Document Template creation page.

In the template, while you create, select "ANZ PEPPOL Custom Data Source Plug-in" from the Custom Data Source Plug-in Implementation drop down.

Limitation:

In the ANZ PEPPOL Template, we currently do not support the population of "ID" under "TaxCategory" and "TaxExemptionReasonCode" values.

The ANZ PEPPOL Template mappings are according to this below table:

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
1	ubl:Invoice			Invoice	This is the root element of the invoice's PEPPOL template. It has a default value.
2	ubl: Credit Note			Credit Memo	This is the root element of the credit memo's PEPPOL template. It has a default value.
3	cbc:CustomizationID			Both	This element identifies the specifications of the rules for semantic content, cardinalities, and business to which the data in the instance document conforms to. This element has a default value.
4	cbc: ProfileID			Both	This element identifies the context of the business process related to the transaction and lets the buyer process the invoice accurately. This element has a default value.
5	cbc:ID	transaction.tranid	Invoice Number or Credit Number	Both	This element displays a unique number that identifies if the transaction is an Invoice or Credit Memo. This value is mapped with the entry number of the transaction.
6	cbc: Issue Date	transaction. trandate	Date	Both	This element displays the issue date value present in the **Date** field of the transaction. The date format is YYYY-MM-DD in a template.
7	cbc:Due Date	transaction. duedate	Due Date	Invoice	This element displays the payment due date value present in the **Due Date** field of an Invoice. The date format is YYYY-MM-DD in a template.
8	cbc:InvoiceType Code		Custom Field	Invoice	This element displays a code specifying the invoice type. This has a default assigned value of 380. You must map the value with the required custom field in the SuiteApp. The following list has information about the Invoice list and the Invoice Type Codes (UNCL1001): ▪ Commercial Invoice – 380 ▪ Consignment Invoice – 395 ▪ Debit Note for Financial Adjustments – 84 ▪ Debit Note for Goods or Services – 80 ▪ Debit Note – 383 ▪ Factored Invoice – 393 ▪ Forwarder's Invoice – 623 ▪ Insurer's Invoice – 575 ▪ Metered Services Invoice – 82 ▪ Prepayment Invoice – 386
9	cbc: Credit Note Type Code		Custom Field	Credit Memo	This element displays a code specifying the credit memo type. This has a default assigned value of 381. You can map the value with the required custom field in the SuiteApp. The following list has the information about Credit Memo list and the Credit Memo Type Codes (UNCL1001): ▪ Credit Note for Goods or Services – 81 ▪ Credit Note for Financial Adjustments – 83 ▪ Credit Note – 381 ▪ Factored Credit Note – 396 ▪ Forwarder's Credit Note – 532
10	cbc:Note	transaction. memo	Memo or Custom Field	Both	This element displays a note with information about any change in the invoice or credit memo. The note is retrieved from the value in the **Memo** field of the transaction. In some SuiteApps, this value can be a combination of the value from the **Memo** field and the value from the custom field on a subsidiary record. You can map the value with the required custom field in the SuiteApp.
11	cbc:Tax Point Date	transaction.taxpointdate	Tax Point Date	Both	This element displays the **Tax Point Date** field value on the Tax Details tab of a transaction.It is available only in accounts with SuiteTax enabled.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					This value represents the date when the GST becomes accountable for the Seller and for the Buyer in so far as that date can be determined and differs from the date of issue of the invoice, according to the GST directive. This element is required if the Value added tax point date is different from the Invoice issue date. The format in which it be displayed is "YYYY-MM-DD". This element is not supported in the ANZ PEPPOL TEMPLATE
12	cbc:DocumentCurrencyCode	transaction.currency.symbol	Currency (Under Subsidiary Record)	Both	This element displays the ISO Code of the currency used in the transaction, if Multi-Currency feature is enabled in an account. The value is not displayed if the Multi-Currency feature is not enabled in an account.
13	cbc:TaxCurrencyCode	custom.subCurrencyISOCode Refer Custom Data Source (CDS) Plug-in Implementation	Currency — In Subsidiary Record	Both	This element displays the ISO Code used by the subsidiary of the transaction if the currency is not same for transaction and the subsidiary. The value of this element is retrieved from the custom data source plug-in used in the template.
14	cbc:AccountingCost		Custom Field	Both	This element displays the details of the booking data related to the buyer's financial accounts. You must map this field value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
15	cbc:BuyerReference	transaction.otherrefnum	PO #	Both	This element displays the identifier or code assigned by the buyer for internal routing. The **PO#** field is used to identify the buyer and the order details. The value can be the name of the person ordering the products, employee number, or an identification code for a buyer, department, or group. You can map the value of this field with the required custom field in the SuiteApp.
16	cac:InvoicePeriod			Invoice	This is a parent element containing a group of business terms that provide information about the invoice period, also known as the delivery period. If you use this group, you must also use the invoice period start date, end date, or both. This element is not supported in the ANZ PEPPOL TEMPLATE
17	cbc:StartDate	transaction.startdate	Start Date	Invoice	This element displays the start date of an invoice period for a transaction. This element is displayed along with a value on the generated e-document only if the value is entered in the **Start Date** field of the transaction. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
18	cbc:EndDate	transaction.enddate	End Date	Invoice	This element displays the end date of an invoice period for a transaction. This element is displayed along with a value on the generated E-Document only if the value is entered in the **End Date** field of the transaction. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
19	cbc:DescriptionCode		Custom Field	Both	This element displays the code of the date when VAT becomes accountable for seller and buyer. For example: ■ 3- Invoice document issue date, time ■ 35- Delivery date, time ■ 432- Paid to date You can map the value with the required custom field in your account. This element is not supported in the ANZ PEPPOL TEMPLATE
20	cac:OrderReference			Both	This is a parent element with details of the order and sales order references.
21	cbc:ID	transaction.otherrefnum customer.accountnumber customer.entityid	1. PO# If 1. is absent, then Account's value on customer record. If 2. is absent, then Customer	Both	This element displays one the following identifiers for a referenced purchase order allotted by the buyer: ■ PO# ■ Account Number – The account number is displayed if the PO# value is missing. ■ Customer ID – The customer ID is displayed if the account number is missing.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
			ID's value on customer record.		The element represents an identifier for a referenced purchase order issued by the Buyer.
22	cbc:SalesOrderID	transaction.createdfrom	Created From	Invoice	This element displays an identifier for a referenced sales order allotted by the seller. It has the entry number of the sales order transaction only if an invoice is created from a sales order. The entry number is retrieved from the **Created From** field in the invoice. If the invoice is not created from a sales order, then the element is not printed.
23	cac:BillingReference			Credit Memo	This is a parent element which contains a group of business terms providing information about invoices to which the credit memo is applied to. The element with all its child elements is created separately for each invoice the credit memo is applied to. This does not contain any value but has child elements. In the template, we can find the element only with no value bound to it. If the credit memo is not applied to any invoice, the element with all its child elements will not be displayed.
24	cac:InvoiceDocumentReference			Credit Memo	This parent element has no value in the template but contains child elements identifying the invoice. In this template, element is not provided with any value.
25	cbc:ID	invoice.invoiceId Refer CDS		Credit Memo	This key displays a value identifying an invoice that was previously sent by the seller, to which the credit memo is applied to. The list of invoices the credit memo is applied to is displayed in the Apply subtab and this key refers to the Reference Number for the particular invoice The value of this field is retrieved from the Custom Data Source Plug-in (CDS) used in the template.
26	cbc: Issue Date	invoice. invoiceDate Refer CDS		Credit Memo	This element is used to show the Issue Date for each invoice to which the credit memo is applied to. The list of invoices the credit memo is applied to is displayed in the Apply Subtab and this key refers to the transaction date for the particular invoice. The format in which it be displayed is "YYYY-MM-DD". The value of this field is retrieved from the Custom Data Source Plug-in (CDS) used in the template.
27	cac:DespatchDocumentReference			Both	This parent element provides reference to the dispatch advice. This element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE
28	cbc:ID		Custom Field	Both	This element refers to an identifier of a referenced dispatch advice. You can map the value of this key with the preferred custom field in your account. This element is not supported in the ANZ PEPPOL TEMPLATE
29	cac:ReceiptDocumentReference			Both	This parent element has reference to the receipt advice. This element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE
30	cbc:ID		Custom Field	Both	This element displays an identifier of a referenced receiving advice. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
31	cac:OriginatorDocumentReference			Both	This parent element contains a tender or lot reference details. This element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE
32	cbc:ID		Custom Field	Both	This element displays the call for tender or lot of the invoice ID. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
33	cac:ContractDocumentReference			Both	This parent element provides details of any existing contract associated with a transaction. This element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE
34	cbc:ID		Custom Field	Both	This element has a value that identifies a contract. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
35	cac:AdditionalDocumentReference			Credit Memo	This parent element has no value, but it has child elements with values associated to it in the template. It has information about group of business terms and details, with additional supporting documents having terms of the invoice. The supporting documents include: ▪ Reference to a document number known to the receiver ▪ External documents referenced by a URL ▪ Embedded documents ▪ Base64 encoded documents like time report Basically, the project reference details are being used in this parent's child element. By default, the element along with all its child elements will only be displayed when the header project is specified.
36	cbc:ID	transaction.job	Project	Credit Memo	This element displays the identifier of an object based on the invoice provided by the seller or a value used for identifying a supporting document. You can map the value with the required custom field in the SuiteApp. By default, the value for this key is used for the identification of the header project if a header project is specified else element is not displayed.
37	@schemeID		Custom value from list	Credit Memo	This element must have a value displaying the identifier of the identification scheme of an invoiced object. It is used inside the cbc:ID element. The value of this element must be from the list. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
38	cbc:DocumentTypeCode		Custom Field	Credit Memo	This element displays the code value as 50 if there is a project specified in the header of the credit memo. If the credit memo has an applied invoice or it does not display any content if conditions are not met, then the code value is 130 instead of 50. Another reason to use the code 130 is when the credit memo does not display any content if the conditions are not met. You can map the value with the required custom field in the SuiteApp. By default, a value of 50 is provided if a header project is specified else the element is not displayed.
39	cbc:Document Description		Custom Field	Both	This element has a value that describes supporting documents such as time sheets and usage reports. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
40	cac:Attachment			Credit Memo	This parent element has information about attached documents with the transaction. The element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE
41	cbc:EmbeddedDocumentBinaryObject		Custom Field	Credit Memo	This element has a value referencing any attached document embedded as a binary object (Base64) or sent together with the invoice. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
42	@mimeCode		Custom Value	Credit Memo	This is a value used with the element cbc:EmbeddedDocumentBinaryObject that represents the mime code of an attached document. For example: text/csv You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
43	@fileName		Custom Value	Credit Memo	This is a value used with the element cbc:EmbeddedDocumentBinaryObject that represents the file name of an attached document. For example: abc.csv You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
44	cac:ExternalReference			Both	This is a parent element providing details of the external document's reference such as its location. The element has no value, but it has child elements with values associated to it in the template.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
45	cbc:URI		Custom Field	Both	The value of this element represents the Uniform Resource Locator (URL) that identifies the location of the external document. This URL locates the resource and its primary access mechanism, such as http:// or ftp://. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE
46	cac:ProjectReference			Invoice	This parent element has information about any project associated with an invoice. This element has no value, but it has child elements with values associated to it in the template. This element is not displayed if its child element do not have any value. You can map the value with the required custom field in the SuiteApp.
47	cbc:ID	transaction.job	Project	Invoice	This element indicates the ID of the project associated with an invoice. If the **Project** field has no value in the transaction, the element is not printed.
48	cac:AccountingSupplierParty			Both	This is a parent element that provides information about the seller using various child elements. This element has no value, but it has child elements with values associated to it in the template.
49	cac:Party			Both	This is a parent element and a child element of cac:AccountingSupplierParty, having information about the seller like electronic address. This element has no value, but it has child elements with values associated to it in the template.
50	cbc:EndpointID	By Default: custom.subTaxRegNo Refer CDS	Custom Field can be used. OR By Default: For Legacy: GST REGISTRATION NO/ ABN (subsidary record) For SuiteTax: SUBSIDIARY TAX REG. NUMBER (Transaction record under taxdetails tab)	Both	This element identifies the seller's electronic address to which the application level response to the invoice can be delivered. The element uses the custom data source plug-in to get one of the following values: In OW Account: ▪ For Legacy: using subsidiary record's federalidnumber. ▪ For SuiteTax Account: using transaction's subsidiarytaxregnum This element is populated only if value exists. In SI Account: This element will not be displayed. You can map the value with the required custom field in the SuiteApp.
51	@schemeID	By Default: custom.schemeId Refer CDS	Custom Value	Both	This element is used with the element cbc:EndpointID and identifies the scheme ID of the seller's electronic address. You can find the scheme ID value from Scheme ID List. Possible values for scheme ID in case of ANZ PEPPOL Template are: New Zealand: 0088 Australia: 0151 The value of this field is retrieved from the Custom Data Source Plug in (CDS) used in the template only for OW Accounts.
52	cac:PartyIdentification			Both	This parent element identifies the seller or the seller's unique banking reference ID provided by the bank. This element has no value, but it has child elements with values associated to it in the template.
53	cbc:ID	By Default: custom.subTaxRegNo Refer CDS	Custom Field can be used. OR By Default: For Legacy: GST REGISTRATION NO/ ABN (subsidary record) For SuiteTax: SUBSIDIARY TAX REG. NUMBER	Both	This parent element identifies the seller or the seller's unique banking reference ID provided by the bank. ICD code list is used for the seller's ID. SEPA is the code used for SEPA bank's assigned creditor reference. For the buyer to identify a supplier automatically, the following identifiers are available: ▪ BT-29 – Seller identifier ▪ BT-30 – Seller legal registration identifier ▪ BT-31 – Seller VAT identifier The element uses the custom data source plug-in to get one of the following values: In OW Account: ▪ For Legacy: using subsidiary record's federalidnumber. ▪ For SuiteTax Account: using transaction's subsidiarytaxregnum

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
			(Transaction record under taxdetails tab)		This element is populated only if value exists. In SI Account: This element will not be displayed. You can map the value with the required custom field in the SuiteApp. You can map the value with the required custom field in the SuiteApp.
54	@schemeID		Custom Value	Both	This element is used with the element cbc:ID to identify the scheme of the seller Identifier. For example, for a bank assigned credit identifier BT-90, the seller ID must be SEPA.
55	cac:PartyName			Both	This parent element displays the seller's information such as name and address. This element has no value, but it has child elements with values associated to it in the template. This element is not displayed if its child elements do not have values.
56	cbc:Name	transaction.subsidiary.name	Addressee (Subsidiary's main address's addressee name) If 1. is absent, use subsidiary name.	Both	This element displays the business name, another name by which the Seller is known, other than the Seller name. It contains a value in the **Addressee** field of the Subsidiary's main address. Example: Parent Company, Arizona Company etc
57	cac:PostalAddress			Both	This is a parent element with information about the seller's address. Required fields of the address must be filled out to comply with the legal requirements. This element has no value, but it has child elements that have values associated to it in the template.
58	cbc:StreetName	transaction.subsidiary.address1	Address 1 (Subsidiary's main address's address line 1)	Both	This element displays the main address line of the seller's address. The element has the value of the **Address 1** field of the subsidiary's main address. This element is displayed only if a value is entered in the **Address 1** field on the Subsidiary record; otherwise, the element is not displayed.
59	cbc:AdditionalStreetName	transaction.subsidiary.address2	Address 2 (Subsidiary's main address's address line 2)	Both	This element displays another address line with more details about the main address. This element has the value of the **Address 2** field in the subsidiary's main address. This element is displayed only if a value is entered in the **Address 2** field on the Subsidiary record; otherwise, the element is not displayed.
60	cbc:CityName	transaction.subsidiary.city	City (Subsidiary's main address's city)	Both	This element displays the common name of the city, town, or village of the seller's location. The element's value is displayed in the **City** field of the subsidiary's main address. If the field value in subsidiary is blank, the element is not displayed.
61	cbc:PostalZone	transaction.subsidiary.zip	ZIP(Subsidiary's main address's zip)	Both	This element identifies a group of properties with addresses based on the relevant postal service. This element is displayed only if there is a value in the **Zip** field of the subsidiary's main address. If the field value in subsidiary is blank, the element is not displayed.
62	cbc:CountrySubentity	custom.mainAddressState Refer CDS	State (Subsidiary's main address's state)	Both	This element identifies the subdivision of the country in the subsidiary's main address. The element is displayed only if there is a value in the **State** field of the subsidiary's main address. If the field value in subsidiary is blank, the element is not displayed. The state in Address field is displayed with codes for Australian States. The code is converted to the State Name when displayed in the template. The mapping for state code and state name are given below: ■ "ACT": "Australian Capital Territory", "NSW": "New South Wales" "NT": "Northern Territory" "QLD": "Queensland" "SA": "South Australia" "TAS": "Tasmania" "VIC": "Victoria" "WA": "Western Australia"
63	cac:AddressLine			Both	This parent element has information about the additional address line in the subsidiary address. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
64	cbc:Line		Custom field	Both	This element has information about an additional address line with more details about the main address line. No value has been provided for the element in this template. This element is not supported in the ANZ PEPPOL TEMPLATE.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
65	cac:Country			Both	This parent element has information about the country associated with the subsidiary's address. This element has no value in the template.
66	cbc:IdentificationCode	Subsidiary country If not, then display the company information country.	custom.subPrimaryCountry (Refer CDS) custom.countryInCompanyInfoCode (Refer CDS)	Both	This parent element displays the ISO Code of the nexus country. It uses the **Custom Data Source plug-in** implemented in the template to display the values. In OW account, the subsidiary's country name is populated. In SI account, the company information country is populated. If the values are not present, the element is not populated.
67	cac:PartyTaxScheme			Both	This parent element with its child element displays the seller's VAT identifier or tax registration details using different transaction fields. This element has no value in the template, but it has child elements that have values associated with them.
68	cbc:CompanyID	custom.subVatRegNo (Refer CDS)	For Legacy Tax: GST REGISTRATION NO. (Transaction's subsidiary) For SuiteTax: EMPLOYEE IDENTIFICATION NUMBER (Transaction's subsidiary)	Both	This element displays the seller's GST identification number or local identification for tax purpose or reference. This enables a seller to indicate their registered tax status. The element displays the value in the **Employee Identification Number** field of a subsidiary in a transaction in the generated e-document. The value is displayed using the Custom Data Source plug-in implemented in the template.
69	cac:TaxScheme			Both	This parent element has a child element cac:PartyTaxScheme that displays the seller's GST identification or tax registration details using various transaction fields. This element has no value, but it has child elements that have values associated to it in the template.
70	cbc:ID		Fixed Value: GST	Both	This element is displayed if the seller is a GST identifier or TAX Identifier. The default value is GST.
71	cac:PartyLegalEntity			Both	This parent element has information about the seller's legal details like registration name. This element has no value, but it has child elements that have values associated to it in the template.
72	cbc:RegistrationName	transaction.subsidiary.legalname	Legal Name	Both	This element displays the seller's registered name as per the following conventions: - National Registry of Legal Entities - Taxable person - Trading as a person or persons The value of the element is retrieved from the subsidiary's **Legal Name** field in a transaction.
73	cbc:CompanyID	Custom Field can be used. OR By Default: For Legacy: GST REGISTRATION NO/ ABN (subsidary record) For SuiteTax: SUBSIDIARY TAX REG. NUMBER (Transaction record under taxdetails tab)	By Default: custom.subTaxRegNo Refer CDS	Both	This element is used as an identifier allotted by an official registrar. It identifies the seller as legal entity or person. The value of this field is retrieved from the Custom Data Source Plug-in (CDS) used in the template. In OW Account: For Legacy: using subsidiary record's federalidnumber For SuiteTax Account: using transaction's subsidiarytaxregnum This element is populated only if value exists. In SI Account: This element will not be displayed. You can map the value with the required custom field in the SuiteApp.
74	@schemeID		Custom Value	Both	This element is used with the element cbc:CompanyID to identify the scheme identifier of a seller's legal registration. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
75	cbc:CompanyLegalForm			Both	This parent element has details about additional legal information related to the seller. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE
76	cac:Contact			Both	This is a parent element and a child of cbc:CompanyLegalForm with information on a group of business terms providing contact information of the seller. No value has

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					been provided to this element, but it has child elements which have values associated to them. This element will only be displayed if Sales Rep of the transaction has either phone number or email address.
77	cbc: Name	transaction. salesrep	Sales Rep	Both	This element provides contact details for a legal entity or representative and displays the **Sales Rep** field value in a transaction. The element is displayed only if there is a value in the **Sales Rep** field of the transaction.
78	cbc:Telephone	transaction. salesrep.phone	Phone	Both	This element displays the phone number of the contact person. It uses the phone number of the sales representative selected in the transaction as the value. This element is displayed only if there is a value in the sales representative's **Phone** field in the employee record.
79	cbc:ElectronicMail	transaction. salesrep.email	Email	Both	This element displays the email address of the contact person and uses the email address of the sales representative selected in the transaction as the value. This element is displayed only if there is value in the sales representative's **Email** field in the employee record.
80	cac:AccountingCustomerParty			Both	This parent element contains a group of business terms, with information about the buyer. This element has no value, but it has child elements that have values associated to it in the template.
81	cac:Party			Both	This is a parent element and a child of cac:AccountingCustomerParty, providing information about the buyer. This element has no value, but it has child elements with values associated in the template.
82	cbc:EndpointID	For Legacy Tax: customer.vat regnumber in customer record For SuiteTax: customer.defaulttaxreg in customer record	Custom Field or In Legacy Tax Account: **Tax Reg Number** field in the customer record In SuiteTax account: **Default Tax Reg** field in the customer record	Both	This element displays a value identifying a buyer's electronic address to deliver the invoice. For Legacy: using customer record's vatregnumber For SuiteTax Account: using customer record's defaulttaxreg Example: 987654321
83	@schemeID		Custom Value	Both	This element is used with the element cbc:EndpointID and has no value in the template. This element identifies the scheme identifier of a buyer's electronic address. Example: 0192 You can map the value with the required custom field in the SuiteApp.
84	cac:PartyIdentification			Both	This is a parent element providing information about the buyer identifier. This element has no value, but it has child elements that have values associated with them.
85	cbc:ID	customer.accountnumber	Account (Customer's transaction)	Both	This element populates the customer identifier. The value of **Account** field on the transaction's customer record is used as value for this element.
86	@schemeID		Custom Value	Both	This element is used with the element cbc:ID to display or populate the scheme ID of the buyer identifier. You can map the value with the required custom field in the SuiteApp.
87	cac:PartyName			Both	This parent element displays the buyer's details. This element has no value, but it has child elements that have values associated with it in the template.
88	cbc: Name	customer.companyname	Company Name	Both	This element displays another name that the buyer is known, other than the Business name. The value in the **Company Name** field on a customer's transaction is displayed in the generated e-document for this element.
89	cac:PostalAddress			Both	This parent element has a group of business terms providing information about the buyer's postal address. This element has no value, but it has child elements that have values associated with it in the template.
90	cbc: StreetName	transaction. billaddr1	Address 1 (Transaction's billing address)	Both	This element displays the main address line of an address. The element retrieves the value of the **Address 1** field from the billing address in a transaction. This element is displayed only if the **Address 1** field has a value.
91	cbc:AdditionalStreetName	transaction. billaddr2	Address 2 (Transaction's billing address)	Both	This element displays an additional address line to have more details on the main address. This element retrieves the value of the **Address2** field from the billing address in a transaction.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
92	cbc:CityName	transaction.billcity	City (Transaction's billing address)	Both	This element displays the common name of the city, town, or village of the buyer's location. This element is displayed only if the **City** field of the transaction's billing address has a value.
93	cbc:PostalZone	transaction.billzip	Zip (Transaction's billing address)	Both	This element identifies a group of properties with address based on the relevant postal service. **Zip** field of the transaction's billing address is used as value for this element. This element is not displayed if this field has no value.
94	cbc:CountrySubentity	transaction.billstate	State (Transaction's billing address)	Both	This element identifies the subdivision of the country. This element is displayed only if the **State** field of the transaction's billing address has a value.
95	cac:AddressLine			Both	This parent element shows information about the additional address line in a transaction's billing address. This element has no value. This element is not supported in the ANZ PEPPOL TEMPLATE.
96	cbc:Line	transaction.billaddr3	Address 3 (Transaction's billing address)	Both	This element displays an additional address line to have more details on the main address line. This element does not contain any value. This element is not supported in the ANZ PEPPOL TEMPLATE.
97	cac:Country				This parent element provides information about the billing address's country in a transaction. This element has no value, but it has child elements that have values associated with it in the template.
98	cbc:IdentificationCode	custom.billCountryISOCode	Country (Transaction's billing address)	Both	This element displays a code identifying the billing country of a transaction. The element displays the ISO code of the country. The value of this element is displayed using the Custom Data Source plug-in implemented in the template.
99	cac:PartyTaxScheme				This parent element has information about the party GST identifier. This element has no value, but it has child elements that have values associated to it in the template.
100	cbc:CompanyID	For Legacy Tax: customer.vatregnumber For SuiteTax: customer.defaulttaxreg	Custom Field or By default: In Legacy Tax Account: Tax Reg Number In SuiteTax account: Default Tax Reg	Both	This element displays the buyer's GST identification number. The field value of transaction's Customer Tax Reg. Number in Suite Tax is used as value for this element. For Legacy Tax, you can use the **Tax Reg Number** field in a customer record as value for this element. You can map the value with the required custom field in the SuiteApp.
101	cac:TaxScheme			Both	This parent element provides information about the tax scheme used by the customer. This element has no value, but it has child elements that have values associated to it in the template.
102	cbc:ID		Custom Field	Both	This element provides the code of the tax scheme being used. This element has a fixed value GST.
103	cac:PartyLegalEntity			Both	This parent element provides legal information about the buyer. This element has no value, but it has child elements that have values associated to it in the template.
104	cbc:RegistrationName	transaction.billaddressee customer.companyname	Addressee Company Name	Both	This element displays the buyer's full name. If the **Addressee** field has a value in the Billing Address of a transaction, then the full name is displayed along with the element. If not, then the element displays the customer's company name.
105	cbc:CompanyID	For Legacy Tax: customer.vatregnumber For SuiteTax: customer.defaulttaxreg	Custom Field or By default: In Legacy Tax Account: Tax Reg Number In SuiteTax account: Default Tax Reg	Both	This element displays an identifier of the buyer as a legal entity or person. This identifier is issued by an official registrar. This element has no value in the template. For Legacy: Using customer record's vatregnumber For SuiteTax Account: using customer record's defaulttaxreg You can map the value with the required custom field in the SuiteApp.
106	@schemeID		Custom Value	Both	This element displays the scheme ID of the buyer's legal registration identifier. This element has no value associated with it in the template.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					You can map the value with the required custom field in the SuiteApp.
107	cac:Contact				This is parent element has multiple child elements and provides contact information relevant to the buyer. This element has no value associated with it in the template.
108	cbc:Name	custom.primary Contact Refer CDS	Customer Primary Contact NAME (customer record)	Both	This element has the point of contact of the buyer's legal entity or person. The value of this field is retrieved from the Custom Data Source Plug-in (CDS) used in the template. Use Primary Contact Name of Customer Record if present, else populate the value with UNDEFINED.
109	cbc:Telephone	custom.primary Contact Refer CDS	Customer Primary Contact PHONE (customer record)	Both	This element displays the phone number of the buyer's contact. This element is displayed in the generated e-document only if the customer's **Phone Number** field has a value. The value of this field is retrieved from the Custom Data Source Plug-in (CDS) used in the template.
110	cbc:ElectronicMail	custom.primary Contact Refer CDS	Customer Primary Contact EMAIL (customer record)	Both	This element displays the e-mail address of the buyer's contact. This element is displayed in the generated e-document only if the customer's **E-mail Address** field has a value. The value of this field is retrieved from the Custom Data Source Plug-in (CDS) used in the template.
111	cac:Payee Party			Both	This parent element has multiple child elements, with information about the payee. This element is used when the payee is different from the seller. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
112	cac:PartyIdentification			Both	This is a parent element and a child of cac:PayeeParty, which has child elements used for identifying the payee party. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
113	cbc:ID		Custom Field	Both	This element is used to identify both the payee and the unique banking reference identifier of the payee assigned by the payee's bank. For identifying the payee, you can use the ICD code list. You can use SEPA code for identifying the SEPA bank assigned creditor reference. For example, FR932874294. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
114	@schemeID		Custom Value	Both	This element used with the cbc:ID element identifies the payee's scheme ID. This has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
115	cac:PartyName				This is a parent element and a child of cac:PayeeParty element. This element provides information about the payee name. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
116	cbc:Name		Custom Field	Both	This element displays the name of the payee. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
117	cac:PartyLegalEntity			Both	This parent element has legal information about the payee. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
118	cbc:CompanyID		Custom Field	Both	This element identifies the payee's legal registration. The identifier is issued by an official registrar and identifies the payee as a legal entity or person. For example, FR932874294. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
119	@schemeID		Custom Value	Both	This element is used with cbc:CompanyID and has the scheme ID of the payee's legal registration identifier. For example: 0002

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
120	cac:TaxRepresentativeParty			Both	This parent element has information about the seller's tax representative. This element has no value, but it has child elements that have values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
121	cac:PartyName			Both	This is a parent element and a child element of cac:TaxRepresentativeParty. This element displays the name of the seller's tax representative. This element has no value, but it has child elements that have values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
122	cbc:Name		Custom Field	Both	This element displays the full name of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
123	cac:PostalAddress			Both	This is a parent element and a child element of cac:TaxRepresentativeParty. The element has no value, but it has child elements with information about the postal address of the tax representative in template. The required address fields must be filled out to comply with legal requirements. This element is not supported in the ANZ PEPPOL TEMPLATE.
124	cbc:StreetName		Custom Field	Both	This element displays the main address line of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
125	cbc:AdditionalStreetName		Custom Field	Both	This element displays the value of an additional address line of the main address of a seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
126	cbc:CityName		Custom Field	Both	This element displays the common name of the city, town, or village of the tax representative's location. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
127	cbc:PostalZone		Custom Field	Both	This element identifies an addressable group of properties according to the relevant postal service of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
128	cbc:CountrySubentity		Custom Field	Both	This element has information about a country's subdivision of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
129	cac:AddressLine			Both	This element is a parent element and a child of cac:PostalAddress, with information about the additional address line of the seller's tax representative main address. This element has clarifications on the addresses. It has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
130	cbc:Line		Custom Field	Both	This element has information about the additional address line of a main address. It gives information about the seller's tax representative's main address. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
131	cac:Country			Both	This element is a parent element and child of cac:PostalAddress, with information about the country of the address. This element has no value, but it has child elements that have values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
132	cbc:IdentificationCode		Custom Field	Both	This element displays the ISO code of the seller's tax representative country. This element has no a value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
133	cac:PartyTaxScheme			Both	This is a parent element with information about the GST party identifier. This element has no value, but it has child elements with values associated with it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
134	cbc:CompanyID		Custom Field	Both	This element represents the GST identifier of the seller's tax representative. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
135	cac:TaxScheme			Both	This parent element displays the tax scheme used by the tax representative. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
136	cbc:ID		Custom Field	Both	This element displays the name of the tax scheme and has a required value GST, added in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
137	cac:Delivery				This parent element has information about the delivery time and location of the invoiced goods and services. This element has no value in the template, but it has child elements with information about the delivery address in the template.
138	cbc:ActualDeliveryDate	transaction.trandate	Custom Field OR By Default: DATE	Both	This element displays the date when the supply of goods and services is completed. This element is displayed along with the value on the generated e-document only if the value is provided in the **Date** field of the transaction. This date must be in YYYY-MM-DD format. It is populated with transaction date. You can map the value with the required custom field in the SuiteApp.
139	cac:DeliveryLocation			Both	This element has information about the location where goods and services were delivered. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
140	cbc:ID		Custom Field	Both	This element displays the identifier for the delivery location of the goods and services. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
141	@schemeID		Custom Value	Both	This element is a part of cbc:ID element and displays the scheme ID of the **Deliver to Location** field. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
142	cac:Address			Both	This parent element has information about the address where invoiced goods or services are delivered. This element has no value in the template, but it has child elements that have values associated with it.
143	cbc:StreetName	transaction.shipaddr1	Address 1 (Transaction's address line 1)	Both	This element displays the main address line in the transaction's shipping address. This element has the value of **Address 1** field in the transaction's shipping address. This element is not displayed if the **Address 1** field has no value in the transaction.
144	cbc:AdditionalStreetName	transaction.shipaddr2	Address 2 (Transaction's address line 2)	Both	This element displays the main address line in the transaction's shipping address. This element has the value of the **Address 2** field in the transaction's shipping address. The element and its value are not displayed if the **Address 2** field has no value.
145	cbc:CityName	transaction.shipcity	City (Transaction's shipping address's city)	Both	This element displays the common name of the city, town, or village of the delivery location in the shipping address. This element has the value of the **City** field in the transaction's shipping address. The element and its value are not displayed if the **City** field has no value.
146	cbc:PostalZone	transaction.shipzip	Zip (Transaction's shipping address's zip)	Both	This element displays an identifier for a group of properties based on their postal service. This element has the value of the **Zip** field in the transaction's shipping address. The element and its value are not displayed if the **Zip** field has no value.
147	cbc:CountrySubentity	transaction.shipstate	State (Transaction's shipping address's state)	Both	This element displays the subdivision of a country in the shipping address. This element has the value of the **State** field in the transaction's shipping address. The element and its value are not displayed if the **State** field has no value.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
148	cac:AddressLine			Both	This parent element displays additional information about the shipping address. This element is not supported in the ANZ PEPPOL TEMPLATE.
149	cbc:Line		Custom Field	Both	This is a parent element and a child of cac:AddressLine, with information about the additional line of an address, supporting the primary address. This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
150	cac:Country			Both	This parent element displays the country of the shipping address. This element has no value, but it has child elements with values in the template.
151	cbc:IdentificationCode	transaction.shipcountry	Country (Transaction's shipping addresse's country)	Both	This element displays the ISO code of the country in the transaction's shipping address. It displays the value of the **Country** field in the template.
152	cac:DeliveryParty			Both	This parent element displays the details of the delivery party. This element has no value but has a child element with values associated to it in the template.
153	cac:PartyName			Both	This is a parent element and a child of cac:DeliveryParty, with information about the delivery party's name. This element has no value but has a child elements with values associated to it in the template.
154	cbc:Name	transaction.shipaddressee	Addressee (Transaction's shipping address's country)	Both	This element displays the name of the customer or party whom the goods and services are to be delivered. This element displays the value of the **Addressee** field in the transaction's shipping address.
155	cac:PaymentMeans			Invoice	This parent element provides a group of business terms by using different child elements to give payment information. This element has no value, but it has various child elements that have values associated with it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
156	cbc:PaymentMeansCode		Custom Field	Invoice	This element displays the payment type code for a payment made or to be made. This element has no value in the template. Example: 30 The following list has the details of the Payment Types and it's associated code: ■ Credit Transfer – 30 ■ Direct Debit – 49 ■ 0postgiro – 50 ■ Credit Card – 54 ■ Debit Card – 55 ■ Bankgiro – 56 ■ Sepa Credit Transfer – 58 ■ Sepa Direct Debit – 59 ■ Online Payment Service – 68 ■ Finland – 93, 94, 95. ■ Instrument not defined – 1 ■ National or Regional Clearing – 9 ■ Cash – 10 By Default: ZZZ is populated in the Invoice Template You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
157	@name		Custom Value	Invoice	This element displays cbc:PaymentMeansCode , the name of the mode of payment. This element has no value in the template. Example: Credit Transfer By Default: NETSUITE is populated in Invoice Template You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
158	cbc:PaymentID		Custom Field	Invoice	This element displays a text value for associating the payment and the invoice issued by the seller. This is used for creditors' critical reconciliation information. The value helps the seller assign an incoming payment to the relevant payment process. This element has no value in the template. A custom field with a payment reference can be used for this element. Example value: 432948234234234 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
159	cac:CardAccount			Credit Memo	This parent element represents a group of business terms with information about the card used to make a payment against an issued invoice. If a buyer made a payment using a card such as a credit or debit card, then the information about the Primary Account Number (PAN) is mentioned in the invoice. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
160	cbc:PrimaryAccountNumberID		Custom Field	Credit Memo	This element is used to represent the Payment card primary account number. The Primary Account Number (PAN) of the card used for payment .In accordance with card payments security standards, an invoice should never include a full card primary account number. Example: 1234 The last 4 to 6 digits of the PAN (BT-87) are indicated if payment card information (BG-18) is provided in the invoice. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
161	cbc:NetworkID		Custom field	Credit Memo	This element displays the card network identifier such as VISA, American Express, Card. Example: VISA You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
162	cbc:HolderName		Custom Field	Credit Memo	This element displays the name of the holder of the card. Example: John Doe You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
163	cac:PayeeFinancialAccount			Invoice	This parent element displays a group of business terms specifying the credit transfer payments. This element has no value, but it has various child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
164	cbc:ID		Custom Field	Invoice	This element identifies a payment account. It displays a unique ID of the financial payment account for a payment service provider to which the payment is made, such as IBAN or BBAN. Example: NO99991122222 This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
165	cbc: Name		Custom Field	Invoice	This element displays the payment account name of the service provider to which the payment is made. Example: Payment Account This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
166	cac:FinancialInst			Both	This parent element displays the details of a payment service provider. This element has no value, but it has various child elements with values associated to it in the template.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
	itutionBranch				This element is not supported in the ANZ PEPPOL TEMPLATE.
167	cbc:ID		Custom Field	Both	This element displays an ID for payment service provider where a payment account is located such as BIC or a national clearing code. Identification scheme identifier is not used in this element. Example: 9999 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
168	cac:PaymentMandate			Both	This parent element displays a group of business terms for direct debit transactions. This element has no value, but it has various child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
169	cbc:ID		Custom Field	Both	This element displays a unique identifier assigned by the payee for referencing a direct debit mandate. This is used to notify the buyer of a SEPA direct debit in advance. Example: 123456 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
170	cac:PayerFinancialAccount			Both	This parent element has no value, but it has various child elements with values associated to it in the template. It has information about the payer's financial account. This element is not supported in the ANZ PEPPOL TEMPLATE.
171	cbc:ID		Custom Field	Both	This element displays an ID with information about the account to be debited by direct debit. Example: 12345676543 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
172	cac:PaymentTerms			Invoice	This parent element has information about the payment terms applied to the due amount. This parent element has no value, but it has a child element with value associated with it in the template.
173	cbc:Note	transaction.terms	Terms	Invoice	This parent element displays text describing the payment terms that apply to the amount due for payments. It can also describe applicable penalties. If an amount is due for payment (BT-115), then the payment due date (BT-9) or the payment terms (BT-20) is indicated. This element displays a value in the **Terms** field on a transaction. Example: Net within 30 days
174	cac:AllowanceCharge			Both	This parent element displays a group of business terms with information about applicable allowances of a complete invoice. This element also displays a group of business terms with information about charges and taxes excluding GST for a complete invoice. This element has no value and is displayed in the generated e-document only if there is a Header Discount or Shipping Cost associated with the transaction.
175	cbc:ChargeIndicator		Custom Field	Both	This element displays the charge type used for a transaction. The generated e-document displays the value true when informing about charges and the value false when informing about allowances. Example: ▪ True for shipping and handling. ▪ False for header discounts. You can map the value with the required custom field in the SuiteApp.
176	cbc:AllowanceChargeReasonCode		Custom Field	Both	This element displays the document level allowance code or the allowance charge reason code. You can use the UNCL5189 code from the code list for allowance of a subset. For charges, you can use the UNCL7161code from the code list. The document level allowance reason code and document level allowance reason display the same reason. Example: 95 for Discount SAA for Shipping and Handling This element does not have a value in the template.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					You can map the value with the required custom field in the SuiteApp.
177	cbc:AllowanceChargeReason		Custom Field	Both	This element displays document level allowance reason or charge reason in text format. The document level allowance reason code and the document level allowance reason, display the same allowance reason. This element displays the following values: ■ Shipping and handling for charges ■ Discount for allowances You can map the value with the required custom field in the SuiteApp.
178	cbc:MultiplierFactorNumeric		Custom Field	Both	This element displays the document level allowance or charge percentage. The percentage is used with the document level allowance base amount to calculate the document level or charge amount. the value must be 20. Example: 20 You can map the value with the required custom field in the SuiteApp.
179	cbc:Amount	For Discount: transaction.discounttotal For Shipping and Handling: transaction.altshippingcost+transaction.althandlingcost	For Discount: DISCOUNT ITEM (Summary section of transaction) For Shipping and Handling: Transaction's Shipping + handling cost (transaction record)	Both	This element displays the document level allowance or charge amount without GST. This value must be rounded off to a maximum of 2 decimal. value of Discount Item field for allowances. Example: 200
180	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with the cbc:Amount element, to display the ISO code of a transaction's currency.
181	cbc:BaseAmount	transaction.subtotal	Subtotal	Both	This element displays the base amount along with the document level allowance or charge percentage. It calculates the document level allowance or charge amount. The value is rounded off to 2 decimal places in the template. A value is provided in the **Subtotal** field in the Summary section of a transaction Example value: 200
182	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:BaseAmount element, to display the currency's ISO code used by a transaction.
183	cac:TaxCategory			Both	This parent element displays the tax category and its details. This element has no value, but it has child elements with values associated to it in the template.
184	cbc:ID		Custom Field	Both	This element displays a code to identify the GST category. Example: Value: S In ANZ, the following codes are used: ■ E - Exempt From Tax – Specifies that taxes are not applicable ■ S – Standard Rate – Specifies the standard rate. ■ Z – Zero Rated Goods – Specifies that the goods are at zero rate ■ G- Free Export Item, TAX Not Charged – Specifies that the item is free to export and taxes are not charged. ■ O- Services outside scope of tax Specifies that the goods and services which do not attract GST where invoices are issued by entities who are not registered and not required to be registered for GST in Australia. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
185	cbc:Percent		Custom Field	Both	This element displays the document level allowance or GST charge rate. The GST rate is displayed as the percentage that applies to the document level allowance or charge. The categories are Shippingtaxcode, Handlingtaxcode, etc. Example: 25

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					This element does not have a value in the template. You can map the value with the required custom field in the SuiteApp.
186	cac:TaxScheme			Both	This parent element displays the details of the tax scheme used in a transaction. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
187	cbc:ID		Fixed Value	Both	This element displays the name of the tax scheme used in a transaction. A fixed GST value is provided in the template for this element.
188	cac:TaxTotal			Both	This parent element displays the total tax details including the amounts for a transaction. When a tax currency code is provided, two instances of tax total must be present and only one instance with tax subtotal must be used. This element does not have a value in the template. This element has no value in the template.
189	cbc:TaxAmount	transaction.taxtotal	Tax	Both	This element displays the total GST amount for an invoice or the total GST amount in the accounting currency as required by the seller. This value is rounded off to a maximum of 2 decimal places in the template. The elements and its values are displayed only if: ■ A transaction's tax total is not equal to zero. ■ The transaction's currency is not equal to the subsidiary's currency of the transaction.
190	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxAmount element, to display the ISO code of the currency used by the transaction.
191	cac:TaxSubtotal			Both	This parent element displays a group of business terms with information about GST break down based on different categories, rates and exemption reasons. This element has no value, but it has child elements with values associated with it.
192	cbc:TaxableAmount	taxDetails (Refer CDS)	Sum of amount of all items calculated per unique tax category	Both	This element displays all the taxable amount for each item specific to a unique GST category code. This value must be rounded off to 2 decimal places. You can map the value with the required custom field in the SuiteApp.
193	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxableAmount element to display the ISO code of the transaction's currency.
194	cbc:TaxAmount	taxableAmount ((Using CDS)	sum of tax amount of all items calculated per category	Both	This element displays the total GST amount for each item specific to a unique GST category code. The value is rounded off to a maximum of 2 decimal places in the template. You can map the value with the required custom field in the SuiteApp.
195	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxAmount to display the ISO code of the transaction's currency.
196	cac:TaxCategory			Both	This parent element displays the GST category details. This element has no value, but it has child elements with values associated with it.
197	cbc:ID		Custom Field	Both	This element displays a code to identify the GST category. Example: Value: S In ANZ, the following codes are used: ■ E - Exempt From Tax – Specifies that taxes are not applicable ■ S – Standard Rate – Specifies the standard rate. ■ Z – Zero Rated Goods – Specifies that the goods are at zero rate ■ G - Free Export Item, TAX Not Charged – Specifies that the item is free to export and taxes are not charged. ■ O- Services outside scope of tax Specifies that the goods and services which do not attract GST where invoices are issued by entities who are not registered and not required to be registered for GST in Australia. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
198	cbc:Percent		Custom Field	Both	This element displays the GST category rate for each unique tax code used by items for shipping. The GST rate is displayed as percentage for the applicable VAT category. Example: 25 This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
199	cbc:TaxExemptionReasonCode		Custom Field	Both	This element displays a GST exemption reason code. A coded statement indicates the reason for exempting the GST amount. This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
200	cbc:TaxExemptionReason		Custom Field	Both	This element displays the GST exemption reason text. Example: Exempt This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
201	cac:TaxScheme			Both	This parent element displays the tax scheme. This element has no value, but it has child elements with values associated with it in the template.
202	cbc:ID		Fixed Value	Both	This element displays the name of the tax scheme used in a transaction. A fixed value GST is assigned to this element.
203	cac:LegalMonetaryTotal			Both	This parent element displays the document totals. This element has no values, but it has child elements that provide monetary totals for an invoice or a credit memo in a template.
204	cbc:LineExtensionAmount	transaction.subtotal	Subtotal	Both	This element displays the sum of all the invoice line net amounts in an invoice. The value is rounded off to a maximum of 2 decimal places in the template. This element displays the **Subtotal** field value from the transaction's Summary tab.
205	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:LineExtensionAmount element to display the ISO code of a transaction's currency.
206	cbc:TaxExclusiveAmount	transaction.altshippingcost + transaction.althandlingcost + transaction.subtotal – transaction.discounttotal	Subtotal + Allowances – Discount	Both	This element displays the total amount of an invoice without GST. The value of this element must be rounded off to two decimals. In a template, you must add the transaction's amount in the Subtotal field in the Summary tab with the handling and shipping charges to get a value. Example: 3600.0 You can add extra allowances and change the template to accommodate this element.
207	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxExclusiveAmount to display the ISO code of the transaction's currency.
208	cbc:TaxInclusiveAmount	transaction.total	Total	Both	This element displays the total amount of a transaction with GST. This value is rounded off to 2 decimal places in the template.
209	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:TaxInclusiveAmount element to display the ISO code of the transaction's currency
210	cbc:AllowanceTotalAmount	transaction.discounttotal	Discount Item	Both	This element displays the sum of allowances at the document level. The value of this element is rounded off to 2 decimal places. Example: 200.0
211	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:AllowanceTotalAmount element to display the ISO code for the transaction's currency.
212	cbc:ChargeTotalAmount	transaction.altshippingcost+ transaction.althandlingcost	Shipping Cost+ Handling Cost	Both	This element displays the shipping cost charge at the document level in the transaction. The value is rounded off to 2 decimal places in the template. It displays total cost of shipping and handling. Example: 10.0
213	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	The element is used with cbc:ChargeTotalAmount to display the ISO code of the transaction's currency.
214	cbc:PrepaidAmount	transaction.amountpaid	Payments	Both	This element displays the sum of the amount paid in advance, for a transaction. The value is rounded off to a maximum 2 decimal places in the template. This value displays the sum of all the payments made for a transaction.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					Example: 1000.0
215	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PrepaidAmount element to display the ISO code of the transaction's currency.
216	cbc:PayableRoundingAmount		Custom Field	Both	This element displays the rounding amount. The amount is added to the invoice total to round off the amount to be paid. The value is rounded off by maximum two decimals in template. Example: 0.0 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
217	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PayableRoundingAmount element to display the ISO code of the transaction's currency. This element is not supported in the ANZ PEPPOL TEMPLATE.
218	cbc:PayableAmount	For Invoice: transaction.amountremainingtotalbox For Credit Memo: transaction.amountremaining	Amount Due	Both	This element displays the amount due for payment and the outstanding amount that is requested to be paid. The value is rounded off to a maximum 2 decimal places in the template. The **Amount Due** field value in the Summary tab is used for the transaction. Example: 3500.0
219	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PayableAmount element to display the ISO code of the transaction's currency.
220	cac:InvoiceLine			Invoice	This parent element displays the line item components. It also has a group of business terms providing information about individual transaction lines. This element has no value, but it has child elements with values in the template.
221	cac: CreditNoteLine			Credit Memo	This parent element displays the line item components on a Credit Memo. It also has a group of business terms providing information about individual invoice lines for the credit memo. This element has no value, but it has child elements with values associated to it in the template. This element is used only for the Credit Memo transaction type.
222	cbc:ID	item.line	Line Sequence Number	Both	This element displays the transaction line ID for each item. It has a unique identifier for an individual line in the transaction. The line sequence number of a line item is used in the template. Example: 12
223	cbc:Note	transaction.memo	Memo	Both	This element is used in the invoice and credit memo line note. It is a text note with unstructured information relevant to the invoice line. The header level **Memo** field is used as value in the template. Example: New article number 12345
224	cbc:InvoicedQuantity	item.quantity	Quantity	Both	This element displays the value of invoiced quantity of each item in the transaction line. The **Quantity** field on the line item is used as value in the template. Example: 100 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.
225	cbc:CreditedQuantity	item.quantity	Quantity	Credit Memo	This element displays the value of invoiced quantity of each item in the credit note line. If the **Quantity** field for the item has no value, this element will not be printed. This element is used only for the Credit Memo transaction type. Example: 100 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid.
226	@unitCode		Custom Value	Both	This element is used with cbc:InvoicedQuantity element, and displays the invoiced quantity for unit of measure. This unit of measure applies to the invoiced quantity. An alphanumeric code consisting of 2 or 3 letters can be used. Codes for unit of packaging from UNECE Recommendation No. 21 can be used in accordance with the descriptions in the "Intro" section of UN/ECE Recommendation 20, Revision 11 (2015): The 2 character alphanumeric code values in UNECE Recommendation 21 shall be used. To avoid duplication with existing code values in UNECE Recommendation No. 20, each code value from UNECE Recommendation 21 shall be prefixed with an "X", resulting in a 3 alphanumeric code when used as a unit of measure.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					Example: C62 This element has a fixed value "ZZ" in the template. You can map the value with the required custom field in the SuiteApp.
227	cbc:LineExtensionAmount	item.amount	Amount	Both	This element displays the total amount of the invoice line. The total amount includes the line level allowances, charges, and other relevant taxes but excludes VAT. The value is rounded off to a maximum of 2 decimal places. The amount of each element is used as the value in the template. This element is not printed if the **Amount** field has no value for an item. Example: 2145.00
228	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:LineExtensionAmount element to display the ISO code of the transaction's currency.
229	cbc:AccountingCost		Custom Field	Both	This element displays the accounting reference of the invoice line buyer. A text value specifies the location of where to book relevant data in the buyer's financial accounts. Example: 1287:65464 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
230	cac:InvoicePeriod			Invoice	This is parent element displays a group of business terms providing information about the invoice line period. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
231	cbc:StartDate		Custom Field	Invoice	This element displays the invoice line period start date. The format of the invoice line start date is YYYY-MM-DD. Example: 2017-10-05 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
232	cbc:EndDate		Custom Field	Invoice	This element displays the invoice line period end date. The format of the invoice line end date is YYYY-MM-DD. Example: 2017-10-15 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
233	cac:OrderLineReference			Both	This parent element displays the reference of the order line. This element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
234	cbc:LineID		Custom Field	Both	This element displays an object ID of a seller's invoice line. Example: AB12345 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
235	cac:DocumentReference			Both	This parent element displays the details of the line object identifier. This element has no value, but it has child elements with values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
236	cbc:ID		Custom Field	Both	This element displays the seller's invoice line object identifier. Example: AB12345 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
237	@schemeID		Custom Value	Both	This element is used with cbc:LineID element to display the ID of the scheme identifier of an invoice line object. Example: ABZ You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
238	cbc:DocumentTypeCode		Fixed value 130 for Invoice	Both	This element displays the document type code.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
			Fixed Value 130 if Credit Memo is created from Invoice		Code "130" indicates the reference for an invoice object. This code is not used for other documents. In the Invoice template, the code value is always 130. If a credit memo is created from an invoice, the code is always 130 in the credit memo template. Default Value: 130 You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
239	cac:AllowanceCharge			Both	This parent element displays a group of business terms with information about allowances or charges of an individual invoice. This element has no value, but it has child elements with values associated with it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
240	cbc:ChargeIndicator		If line item is of type Discount, display "false" For other type of items, display "true"	Both	This element displays "true" or "false" value for the item type in a transaction. If the item type is discount, the value is "false". For item types other than discount, the value is "true" in the generated e-document. This element is not supported in the ANZ PEPPOL TEMPLATE.
241	cbc:AllowanceChargeReasonCode		If line item is of type Discount, the value 95 is displayed For other type of items, display no value	Both	This element displays the line level allowance or charge reason code. If the line item type is Discount, the value of the code is 95; otherwise, the element will not have a value in the template. Example: 95 This element is not supported in the ANZ PEPPOL TEMPLATE.
242	cbc:AllowanceChargeReason		If item type is Discount, then display "Discount" For other type of items, the element has no value	Both	This element displays the line level allowance or charge reason in text. If the line item type is Discount, the value of the field is **Discount** in template; otherwise, the element is not displayed. Example: Discount This element is not supported in the ANZ PEPPOL TEMPLATE.
243	cbc:MultiplierFactorNumeric	item.rate	If item type is Discount, the values of Unit Price or Rate are displayed. For other type of items, the element has no value - Unit Price (Item) or Rate (Item).	Both	This element displays the line level allowance or charge percentage depending on the line level allowance base amount. It calculates the line level allowance or charge amount only for discount item type. For discount item type, the unit price or rate value of the item is displayed. If item type is not discount, the element is not available in the generated e-document. For discount item type, the unit price or rate value of the item is displayed. If unit price or rate is in percentage, the value will be in units. Example: 20% will be 20 If unit price or rate is not percentage, the value will be in decimals. Example: 20 will be 20.00 This element is not supported in the ANZ PEPPOL TEMPLATE.
244	cbc:Amount	item.amount	If item type is Discount, the Amount value is displayed. For other item types, the element has no value. Amount	Both	This element displays the line level allowance or charge amount without GST. The value is rounded off to maximum two decimals. Example: 200 This element is not supported in the ANZ PEPPOL TEMPLATE.
245	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:Amount element to display the ISO code of the transaction's currency. This element is not supported in the ANZ PEPPOL TEMPLATE.
246	cbc:BaseAmount		Custom Field	Both	This element displays the line level allowance or charge base amount. The base amount is used according to the line level allowance or charge percentage. This is to calculate the line level allowance or charge amount. The value is rounded off to maximum two decimals. Example: 1000 This element is not supported in the ANZ PEPPOL TEMPLATE.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
247	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:BaseAmount element to display the ISO code of the transaction's currency. This element is not supported in the ANZ PEPPOL TEMPLATE.
248	cac:Item			Both	This parent element displays a group of business terms with information about invoiced goods and services. This element has no value, but it has child elements with values associated with it in the template.
249	cbc:Description	item.description	Description	Both	This element displays the description and features of each item in a transaction. The value of the **Description** field for items is displayed. Example: Long description of an item
250	cbc:Name	item.name	Item Name	Both	This element displays the item name entered in the **Item Name** field in a transaction. Example: Item name
251	cac:BuyersItemIdentification			Both	This parent element displays the buyer's item identification and has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
252	cbc:ID		Custom Field	Both	This element displays an identifier assigned by the buyer for an item. Example: 123455 This element has no value in the template. You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
253	cac:SellersItemIdentification			Both	This parent element displays the information about seller's identification. This element has no value, but it has child elements with values associated with it in the template. You can map the value with the required custom field in the SuiteApp.
254	cbc:ID	item.item	Item Name or Item Number	Both	This element displays an identifier assigned by the seller for an item. The value is displayed only for items without discount and description in the generated e-document. Example: 9873242
255	cac:StandardItemIdentification			Both	This parent element has standard item identification details. This element has no value, but it has child elements with values associated with it in the template.
256	cbc:ID	item1.itemUpc Code (Refer CDS)	UPC Code	Both	This element displays the standard item identifier based on a registered scheme. Example: 10986700 This element uses Custom Data Source plug-in to source the value of an item's UPC code in a template.
257	@schemeID		Custom value	Both	This element is used with cbc:ID element, and displays the scheme ID for the item standard identifier. This element has a default value 0160. Example: 0160 This element does not have a value in the template. You can map the value with the required custom field in the SuiteApp.
258	cac:OriginCountry			Both	This parent element is used to display an item's origin country. This element has no value, but it has child elements with values associated with it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
259	cbc:IdentificationCode	item1.item Country	Manufacturer Country	Both	This element displays the ISO code of an item's country of origin. Example: CN This element uses Custom Data Source plug-in implementation to source the value of an item's manufacturer country. This element is not supported in the ANZ PEPPOL TEMPLATE.
260	cac:CommodityClassification			Both	This parent element displays the commodity classification code. This is required when members from same country buy from each other, and the traders need statistical information in the invoice. It is that you use the Item Classification Identifier (BT-158) with the code HS, as an identifier list for this purpose.

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
					This element has no value, but has child elements with values associated with it in the template.
261	cbc:ItemClassificationCode	item1.itemUpcCode	UPC Code	Both	This element displays an item classification identifier code to classify items based on its type or nature. Example: 9873242 This element uses Custom Data Source plug-in to source the value of an item's UPC code.
262	@listID		Fixed Value	Both	This element is used with cbc:ItemClassificationCode element and displays a scheme ID for item classification identifier. It has a fixed value HS. Example: STI You can map the value with the required custom value in the SuiteApp
263	@listVersionID		Custom Value	Both	This element is used with cbc:ItemClassificationCode element and displays scheme ID for the item classification identifier. This element has no value in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
264	cac:ClassifiedTaxCategory			Both	This parent element displays a group of business terms with information about the GST applicable for invoiced goods and services in an invoice line. This element has no value, but it has child elements with values associated with it in the template.
265	cbc:ID		Custom Field	Both	This element displays the GST category code for an invoiced item. Example: S This element has no value in the template. You can map the value with the required custom field in the SuiteApp.
266	cbc:Percent	item.taxrate1	Tax Rate	Both	This element displays the GST rate percentage applied to the invoiced item. It displays the tax rate value of an item. Example: 25
267	cac:TaxScheme			Both	This element displays information about tax scheme. This parent element has no value, but it has child elements with values associated with it in the template.
268	cbc:ID		Fixed Value	Both	This element displays an ID of the tax scheme used in an invoice. A fixed value GST is added in the template.
269	cac:AdditionalItemProperty			Both	This parent element displays a group of business terms with information about item attributes of invoiced goods and services. This element has no value, but it has various child elements that have values associated to it in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
270	cbc: Name		Custom Field	Both	This element displays the name of the item attribute or the property of the item. Example: Color You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
271	cbc: Value		Custom Field	Both	This element displays a value for the item attribute. The value of the attribute or property of the item. Example: Black You can map the value with the required custom field in the SuiteApp. This element is not supported in the ANZ PEPPOL TEMPLATE.
272	cac:Price			Both	This parent element displays the price details. It has child elements with information about the price applied for the invoiced goods and services in an invoice. This element has no value, but it has various child elements with values in the template.
273	cbc:PriceAmount	Item.rate	Unit Price or rate	Both	This element displays the price of an item excluding GST and item price discount. The item net price must be equal to the item gross price, which is less than the item price discount if both the prices are provided. The item price cannot be negative. The unit price or rate of an item is used as a value and must be rounded off to 2 decimal places. Example: 10.8955 is rounded to 11

SL No.	ID	Field ID	Technical Field Name	Applicable for Transaction Type	NetSuite Field Mapping Logic Used for Generic Templates
274	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:PriceAmount to display the ISO code of the transaction's currency.
275	cbc:BaseQuantity	item.quantity	Quantity	Both	This element displays item price base quantity of item units, to which the price applies. Example: 1 The element value with a decimal separator (.) or a whole number is valid. Group separator or more than one decimal separator is invalid. This element is not supported in the ANZ PEPPOL TEMPLATE.
276	@unitCode		Custom Value	Both	This element displays the unit of measure code applicable to the item price base quantity. Also, the unit of measure must be same as the unit code of the invoiced or credited quantity. The unit of measure value has a 3-digit alphanumeric code. This element has no value in the template. Example: C62 This element is not supported in the ANZ PEPPOL TEMPLATE.
277	cac:AllowanceCharge			Both	This element displays allowance details of an invoice. This element has no value, but it has child elements with values in the template. This element is not supported in the ANZ PEPPOL TEMPLATE.
278	cbc:ChargeIndicator		Fixed Value	Both	This element indicates the charge. A charge on price level is not valid. This element has a fixed value false. This element is not supported in the ANZ PEPPOL TEMPLATE.
279	cbc:Amount	item.amount	Amount	Both	This element is used to calculate the item price discount. The total discount is subtracted from the gross item price to calculate the net item price. This element displays the value if only if the item type is discount. Example value: 100 This element is not supported in the ANZ PEPPOL TEMPLATE.
280	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:Amount element to display the ISO code of the transaction's currency. This element is not supported in the ANZ PEPPOL TEMPLATE.
281	cbc:BaseAmount	item.amount	Quantity * Unit Price/Rate = Amount	Both	This parent element displays the gross item price. Item amount is used as value for this element. It is the unit price excluding GST before subtracting the item price discount. The item price discount value must not be negative. The formula for calculating gross item price is, quantity multiplied by unit, and price divided by rate per Item. Alternatively, the item amount can also be used. Example: 123.5 This element is not supported in the ANZ PEPPOL TEMPLATE.
282	@currencyID	transaction.currency.symbol	Transaction's Currency	Both	This element is used with cbc:BaseAmount element to display the ISO code of the transaction's currency. This element is not supported in the ANZ PEPPOL TEMPLATE.

Vendor Bill

Note: To use the templates, paste the content into the "FIELD MAPPING FOR INBOUND E-DOCUMENT" field of the new E-Document Template creation page.

In the template, while you create, select "ANZ PEPPOL Inbound CDS" from the Inbound Custom Data Source Plug-in Implementation drop down.

If no tax rate is present in for the item in XML reference, then by default, tax code with 0% tax rate will be applied to the item in the bill. This is applicable only if one tax code with tax rate 0% is present in the account.

If there are multiple tax codes of same rates or no tax codes present with specified tax rate, then an error will be thrown. The inbound conversion will be failed.

Limitations:

- Expenses is not supported in this template; users can change the template and add expenses, if required.
- Items with vendor code will only be considered for inbound conversion. So, users have to make sure that the items in the XML Reference have vendor code value under the Vendors section of the Purchasing Tab in the item records along with the Vendor's Name. The vendor code should be same as the Item name.
- Inbound Conversion using PEPPOL for Australia and New Zealand is not supported in SuiteTax accounts.
- No other language other than English is supported for Inbound Conversion.
- Multi-Subsidiary vendors are not supported for Inbound Conversion.

S.NO	JSON Key in Template	XML Element applied from XML Reference	Information for the Key
1	tranid	D:Invoice/cbc:ID	This key uniquely identifies the Vendor Bill, that is, Reference Number.
2	trandate	D:Invoice/cbc:IssueDate	This key displays the issue date of the bill provided by the vendor, that is, Date. The date is in the format of the system preferences set in the account. If the element cbc:IssueDate is missing in the XML reference file then it will throw an error "The issue date XPath for the transaction could not be found in the XML reference file." and If the value of the cbc:IssueDate element in XML Reference file is not in the format YYYY-MM-DD then it will throw an error "The issue date of the transaction in the XML reference file is not in correct format.".
3	duedate	D:Invoice/cbc:DueDate	This key displays the payment due date. The date is in the format of the system preferences set in the account. The Bill will be populated only if the XML Reference has relevant value. If the value of the cbc:DueDate element in XML Reference file is not in the format YYYY-MM-DD, you get the error "The due date of the transaction in the XML reference file is in invalid format."

S.NO	JSON Key in Template	XML Element applied from XML Reference	Information for the Key
4	memo	D:Invoice/cbc:Note	This element displays unstructured note related to the Bill. The Bill will be populated only if the XML Reference has relevant value.
5	currency	D:Invoice/cbc:Document CurrencyCode	This key displays the Currency Name in Vendor Bill. The value of this field is derived from the Custom Data Source Plug-in (CDS) used in the template. It retrieves Currency Name from the Currency ISO code present in the XML Reference. Multi Currency feature must be enabled in the account.
6	item	D:Invoice/cac:InvoiceLine	This key iterates over all line items in the XML Reference.
7	vendorcode	D:Invoice/cac:InvoiceLine/cac:Item/cbc:Name	This key displays Vendor Code that is same as the item name. It does not support any other reference apart from item name. Check Limitations section for more details.
8	vendorrate	D:Invoice/cac:InvoiceLine/cac:Item/cbc:Name	This key displays Vendor Name that is same as the item name.
9	quantity	D:Invoice/cac:InvoiceLine/cac:Item/cbc:InvoicedQuantity	This key displays Quantity of each line item. Quantity is not added if the quantity of each line item is not present in the XML Reference. The element value with a decimal separator (.) or a whole number is valid. Group separator (Thousands separator) or more than one decimal separator is invalid.
10	rate	D:Invoice/cac:InvoiceLine/cac:Item/cac:Price/cbc:PriceAmount	This key displays net price or unit price of the item. It displays the rate of each line item, exclusive of GST after subtracting item price discount. The element value with a decimal separator (.) or (,), or a whole number is valid. Group separator (Thousands separator) or more than one decimal separator is invalid.
11	taxcode	D:Invoice/cac:InvoiceLine/cac:Item/cac:ClassifiedTaxCategory/cbc:Percent (We are fetching the tax percent and from that we are finding the corresponding tax code in the account)	This key provides tax codes that is extracted from the tax rate in the XML Reference. [Check the XML Element given in second column]. The tax code which belongs to the specific tax rate for the applicable country. You need to extract the tax codes from this XML Reference: D:Invoice/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode. In OW Accounts, it will retrieve Subsidiary country information. In SI accounts, it will retrieve information about the country from the company information page being populated.

S.NO	JSON Key in Template	XML Element applied from XML Reference	Information for the Key
			1. The following conditions can stop the conversion and throw error: ▪ If you have more than one tax code applicable of **Purchase** or **Both** tax type present for a tax rate in the account. ▪ If no tax code is present for a tax rate. ▪ If the tax code applicable is inactive. ▪ If more than one applicable tax code is present for a tax rate. It throws the below error. "Appropriate tax code could not be found for one of the following reasons: -No such tax code exists. -Multiple tax codes with the same tax rate exists. -Tax code is not of type Purchase or Both. -Tax code is inactive." Rectify the tax codes and make the combination of tax code to tax rate unique for the applicable country. 2. If the tax rate is not present in the XML reference file for a line item, the tax code with tax rate 0% is automatically assigned to that line item. In the below case, following conditions can throw an error: ▪ If more than one applicable tax code with tax rate 0% is present in the account. ▪ If there are no tax codes present for 0%. ▪ If the tax code applicable is inactive. ▪ If not of type Both or Purchase. "Appropriate tax code could not be found for one of the following reasons: -No such tax code exists. -Multiple tax codes with the same tax rate exists. -Tax code is not of type Purchase or Both. -Tax code is inactive." In this case, add a tax code which is active, of type Purchase or Both and of tax rate 0% for the applicable country in the account. 3. For Australia and New Zealand, we are not supporting inbound conversion using PEPPOL in SuiteTax Accounts. If you try to use this template along with CDS in SuiteTax enabled accounts, then conversion will fail and you get the error "Inbound conversion is not supported for SuiteTax accounts in Australia and New Zealand." PEPPOL Tax Category is currently not supported for ANZ. If tax category is

S.NO	JSON Key in Template	XML Element applied from XML Reference	Information for the Key
			present in the inbound XML file, it will not be considered for determining the tax code. The element value with a decimal separator (.) or (,), or a whole number is valid. Group separator (Thousands separator) or more than one decimal separator is invalid.
12	amount	D:Invoice/cac:InvoiceLine/cbc:LineExtensionAmount	The key displays invoice line net amount. The amount is "net" without GST, that is, inclusiveness of line level allowances and charges along with other relevant taxes. The element value with a decimal separator (.) or (,), or a whole number is valid. Group separator (Thousands separator) or more than one decimal separator is invalid.
13	description	D:Invoice/cac:InvoiceLine/cac:Item/cbc:Name	This key's description is taken from the item's name. You can change the reference in the XML Reference if needed.
14	inventiorydetailreq		This key is a required field for items. The default value is set to 'False' for each item.

Understanding Inbound E-Document Templates in JSON Format

Implementing an inbound e-document template enables the system to map which elements in the received XML file will provide data to which fields in the vendor bill record to be created from the XML file.

An inbound e-document template is in JSON format. For more information about JSON objects, go to the w3schools website JSON Introduction.

The bundle also includes a sample JSON template that can be used for parsing XML inbound e-documents for conversion into vendor bills. You can download the sample JSON template from the File Cabinet. The JSON inbound template contains the mapping to basic bill information:

- tranid
- trandate
- currency
- memo
- item
  - vendorname/vendorcode
  - amount
  - rate
  - quantity
  - description
  - tax1amt
- createdfrom

Note: The alias for the inbound XML e-document object is 'XML'. Use this when mapping the XML elements to keys. For example, ${XML.ParentElement.ChildElement}.

You can use or customize the sample JSON template that contains the mapping to basic vendor bill information:

```
{
   "tranid": "${XML.Invoice.InvoiceHeader.InvoiceNumber}",
   "trandate": "${XML.Invoice.InvoiceHeader.InvoiceDate}",
   "currency": "${XML.Invoice.InvoiceHeader.Currency}",
   "memo": "${XML.Invoice.InvoiceHeader.Memo}",
   "createdfrom": "${XML.Invoice.InvoiceHeader.PONumber}",
   "item":[
      <#list XML.Invoice.InvoiceDetails.InvoiceItem as item>
      {
         "vendorcode": "${item.ItemName}",
         "quantity": "${item.Quantity}",
         "rate": "${item.UnitPrice?replace("$", "")}",
         "amount": "${item.LineItemSubtotal?replace("$", "")}",
         "description": "${item.Description}",
         "tax1amt": "${item.TaxAmount?replace("$", "")}"
      }
      <#if item_has_next>,</#if>
          </#list>  ],
   "expense":[
      <#list XML.Invoice.InvoiceDetails.InvoiceExpense as expense>
      {
         "amount": "${expense.Amount?replace("$", "")}",
         "memo": "${expense.Description}"
      }
      <#if expense_has_next>,</#if>
          </#list> ]
}
```

`tranid`, `trandate`, `currency`, `memo`, `item`, expense and `createdfrom` are called key names. Every key name must correspond to a field ID in the vendor bill record to be created from the received XML file. The key name is the reference that points to a field in the vendor bill record. Each key name must extract a value from the received XML file. The value of the key name will be the data that will be entered in the corresponding field of the vendor bill record.

`tranid` is a required key, used as reference number of the vendor bill. `item` is another required key name that is an of JSON objects with details of each item in the vendor bill.

`createdfrom` is a key name used if the vendor bill record to be created is from a Purchase Order. `createdfrom` will take up the value of the PO# of the source purchase order.

`item` is another required key name that is an of JSON objects with details of each item in the vendor bill. Under `item` is `vendorcode`, which is a required key name if the Multiple Vendor feature is enabled. `vendorcode` maps to the code assigned to a specific vendor of an item. If the Multiple Vendor feature is not enabled, the `vendorname` key name must be used. `vendorname` maps to the vendor name/code field of an item.

expense is also an of JSON objects that takes each expense in the vendor bill. Under expense are the `amount` and `memo` key names.

Note: Ensure that your item records are updated and must have unique `vendorname` or `vendorcode`. Also, you must specify the Default Expense Account in the vendor record if you expect to receive bills for expense lines.

At least the required key names must be present in an inbound e-document template. You must not change or edit required key names.

If you have custom records or fields, you can create your own key names that correspond to the field IDs of those custom fields. But make sure that your custom key names have data to extract from the XML files that you will receive from your vendors or other parties.

After setting up the key names and values of your JSON template, you can now implement it as an inbound e-document template. For more information, refer to step 6 of Creating E-Document Templates.

Understanding XSD in Inbound E-Document Templates

The XSD file or XML Schema is a text file that defines and validates what XML elements and attributes must be present in the received XML, before it is uploaded as an inbound e-document record. After the received XML file is validated, the system assigns an e-document template to it. The first matched e-document template will be assigned to the newly created inbound e-document.

The following is a sample XSD that you can use as a reference in creating your own XSD file:

```
<xs:element name="edoc" type="edocType"/>

   <xs:complexType name="edocType">
      <xs:sequence>
         <xs:element name="tranid" type="xs:string"/>
         <xs:element name="po" type="xs:integer"/>
         <xs:element name="memo" type="xs:string"/>
         <xs:any processContents="skip" minOccurs="0"/>
      </xs:sequence>
   <xs:attribute name="version" type="xs:string" use="required" fixed="1.1"/>
</xs:complexType>

</xs:schema>
```

This sample XSD validates an XML document for the following:

- The root element is edoc is present, which must have an attribute `version`, with a value of `1.1`.
- The edoc element must have `tranid`, po, and `memo` elements in the correct order, where:
  - `tranid` is a string
  - po is an integer
  - `memo` is a string

If this sample XSD is used, a received XML file must contain all the elements and attributes defined in the XSD. If it does, the E-Document Template record that the XSD is a part of, will be applied to the received XML file, which is then uploaded as an inbound e-document record. Other elements and attributes can be present in the received XML file in addition to the required ones defined in the XSD. You can define your own required elements and attributes in the XSD file you will create.

Understanding XSD in Outbound E-Document Templates

The XSD file or XML Schema is a text file. It defines and validates what XML elements and attributes must be present in the generated XML, before it is saved as a generated e-document. A detailed audit trail message is logged if the validation fails.

The following is a sample XSD that you can use as a reference in creating your own XSD file:

```
<xs:element name="edoc" type="edocType"/>

   <xs:complexType name="edocType">
      <xs:sequence>
         <xs:element name="tranid" type="xs:string"/>
         <xs:element name="po" type="xs:integer"/>
```

```
        <xs:element name="memo" type="xs:string"/>
        <xs:any processContents="skip" minOccurs="0"/>
      </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required" fixed="1.1"/>
</xs:complexType>

</xs:schema>
```

This sample XSD validates an XML document for the following:

- The root element `edoc` is present, which must have an attribute `version`, with a value of `1.1`.
- The `edoc` element must have `tranid`, `po`, and `memo` elements in the correct order, where:
  - `tranid` is a string
  - `po` is an integer
  - `memo` is a string

If this sample XSD is used, the generated XML file must contain all the elements and attributes defined in the XSD. If it does, the e-document is successfully generated. Other elements and attributes can be present in the generated XML file in addition to the required ones defined in the XSD. You can define your own required elements and attributes in the XSD file you will create.

The **Outbound XSD File** field is used for validating the generated e-document, only if you select the XML format from the **Content Type** dropdown list.

Creating a Digital Signature Plug-in Implementation for E-Documents

A Digital Signature plug-in implementation for e-documents will enable you to generate digitally signed e-documents. You can select a digital signature plug-in from the **Digital signature Plug-in Implementation** field in an e-document template record to digitally sign e-documents. Only the XML or JSON files are generated with a digital signature. The generated PDF files are not digitally signed.

You must first create a custom plug-in implementation for digital signature and then implement it in NetSuite. After this, the plug-in will be available for you to select from the **Digital Signature Plug-in Implementation** field on the e-document template record. To create this plugin implementation, you must first create a JavaScript file for the digital signature plug-in implementation. The JavaScript file must be compatible with SuiteScript 2.0. For more information about creating a Javascript file, refer to the help topic SuiteScript 2.x Script Creation Process.

The plug-in script must return an object with the following function:

```
/**
 * Copyright (c) 2019, Oracle NetSuite  its affiliates. All rights reserved.
 *
 * @NApiVersion 2.x
 * @NModuleScope Public
 * @NScriptType plugintypeimpl
 */

define(["N/crypto/certificate", "N/file"], function(certificate, file) {
    /**
    * @param {Object} pluginContext
    * @param {String} pluginContext.unsignedString
    * @param {String} pluginContext.subsidiaryId
    * @param {String} pluginContext.tranType
    * @param {String} pluginContext.tranId
    * @param {Number} pluginContext.userId
    *
    * @returns {Object} result
    * @returns {string} result.success
```

```
    * @returns {String} result.signedString
    * @returns {String} result.message
    */

     function signDocument(pluginContext) {
         var unsignedString = pluginContext.unsignedString;

         /* Extract the other required values from pluginContext:
                             var subsidiaryId = pluginContext.subsidiaryId;
                             var tranType = pluginContext.tranType;
                             var tranId = pluginContext.tranId;
                             var userId = pluginContext.userId;
         */

         /* Sample params for N/certificate.signXml()
                         var rootTag = "RootTag";
                         var certificateId = "custcertificatesfd";
                         var algorithm = "SHA1";
         */

         var result = {
             success: true,
             signedString: unsignedString,
             message: "This is a sample implementation of Digital Signature.",
         };

         /**
      * Call services to sign the string
      */
         try {
             /*
                                 var signedXML = certificate.signXml({
                                     algorithm : algorithm,
                                     certId : certificateId,
                                     rootTag : rootTag,
                                     xmlString : unsignedString
                                 });

                                 result.success  = true;
                                 result.signedString = signedXML.asString();
                                 result.message = "Document signed successfully";
                             */
         } catch (e) {
             result.success = false;
             result.signedString = "";
             result.message = e.message;
         }

         return result;
     }

     return {
         signDocument: signDocument,
     };
});
```

This script takes the input **pluginContext** which is a JSON object. The parameters of this object are listed in the following table.

Parameter	Type	Description	Remarks
unSignedString	String	This field holds the generated e-document of a transaction in XML or JSON format. The format depends on its selection in the E-Document Template used for generation. The e-document is not digitally signed.	
subsidiaryId	String	This field holds the Subsidiary ID of a transaction.	This parameter can be blank if the subsidiary information is unavailable in the transaction.

tranType	String	This field displays the type of transaction.	tranId along with tranType provides context about the transaction being digitally signed. You can use them together to reference any field value of the transaction using N/search.lookupFields or N/record.load
tranID	String	This field displays the internal ID of the transaction.	
userId	number	This field holds the internal ID of the current logged in user.	This field value can be used wherever there is a requirement to refer to the current logged in user. For example, to update e-document audit trail by shared module API, userId can be used in owner property.

This script must implement digital signature process on an unsigned XML or JSON string. For example Suitescript 2.0 module N/crypto/certificate. This results in a signed XML or JSON string which must be returned in a JSON object with parameters listed in the following table.

Parameter	Type	Description	Required/Optional
success		Valid Values are either true or false. Set the value to true, if the digital signature plug-in implementation is successful and e-document is digitally signed. A string will be generated. Otherwise, set it to false.	Required
signedString	String	This parameter holds the signed string when the digital signature plug—in implementation is successfully completed.	Required
message	String	Message to be passed to Electronic Invoicing. This message gets displayed in E-Document audit trail of the transaction.	Required

Important: The digital signature plug-in implementation script must have the `@NSScriptType plugintypeimpl`.

To create a plug-in implementation record:

1. Go to Customization > Plug-ins > Plug-in Implementations > New.
2. Create a JavaScript file following the sample script.
3. Select the JavaScript file you created, from the **Script File** field and click **Create Plug-in Implementation** button.
4. Select the Digital Signature for E-Document plug–in type from the Select Plug-in Type record.
5. Type the information in the required fields on the Plug-in Implementation record.
6. Click **Save**.

a digital signature plug-in implementation record is created, you can select it from the E-Document Template record. For more information, see the help topics Custom Plug-in Creation and Digital Signing.

Creating an Outbound Validation Plug-in Implementation for E-Documents

An Outbound Validation Plug-in Implementation for e-documents will enable you to validate the e-document during generation process with your custom logic. You can select an Outbound Validation Plug-in Implementation from the **Outbound Validation Plug-in Implementation** field in an E-Document Template.

To use custom validation, you must first create a custom plug-in implementation for outbound validation plug-in for e-documents. After this, the plug-in implementation will be available for you to select in the **Outbound Validation Plug-in Implementation** field on the E-Document Template.

To create this plug-in implementation, you must first create a JavaScript file that must be compatible with SuiteScript 2.0. For more information about creating a Javascript file, see the help topic SuiteScript 2.x Script Creation Process.

The following code is a sample plug-in implementation script for outbound validation:

```
/**
 * @NApiVersion 2.x
 * @NModuleScope Public
 * @NScriptType plugintypeimpl
 */
define([], function() {

    /**
     *
      * @param {Object} pluginContext
      * @param {String} pluginContext.content
      * @param {String} pluginContext.transactionInfo.transactionId
      * @param {String} pluginContext.transactionInfo.transactionType
      * @param {Number} pluginContext.userId
      * @returns {Object} result
      * @returns {string} result.success
      * @returns {String} result.message
      */
     function validate(pluginContext) {

         var result = {
             success: false,
             message: "Validation failed."
         };

         try {

             /**
              * Extract the values from pluginContext
              */

             // var content = pluginContext.content;
             // var userId = pluginContext.userId;
             // Connect to validation service and get response.

             /**
              * Use this information to fetch the transaction data
              */

             /*
              var transactionType = pluginContext.transactionInfo.transactionType;
              var transactionId = pluginContext.transactionInfo.transactionId;
              var transObj = record.load({
                     type: transactionType,
                     id: transactionId
                 });
             */

             // If successful
             result.success = true;
             result.message = "Validation successful!";

             // Sample result if not successful
             // result.success = false;
             // result.message = "Validation failed.";

             return result;

         } catch (e) {
             result.success = false;
             result.message = e.message;
```

```
62.           }
63.
64.           return result;
65.       }
66.
67.       return {
68.           validate: validate
69.       };
70.   });
```

This script takes the input **pluginContext** which is a JSON object. The parameters of this object are listed in the following table.

Parameter	Type	Description	Remarks
content	String	This field holds the generated e-document of a transaction in XML or JSON format. The format depends on its selection in the E-Document Template used for generation.	
transactionInfo. transactionType	String	The type of transaction being validated.	transactionId and transactionType parameters provide information about the transaction being validated. You can use them together to reference any field value of the transaction using N/search.lookupFields or N/record.load
transactionInfo. transactionId	String	The internal ID of the transaction being validated.	
userId	number	This field holds the internal ID of the current logged in user.	This field value can be used wherever there is a requirement to refer to the current logged in user. For example, to update e-document audit trail by shared module API, userId can be used in owner property.

This script can implement any custom validation logic on edocString in the validate function. The validate function should return JSON object with parameters listed in the following table.

Parameter	Type	Description	Required/Optional
success		Valid Values are either true or false. Set the value to true, if the validation is successful. Otherwise, set it to false.	Required
message	String	Message to be passed to Electronic Invoicing. This message gets displayed in E-Document audit trail of the transaction.	Required

Important: The Outbound Validation Plug-in implementation script must have the `@NSScriptType plugintypeimpl`.

To create a plug-in implementation record:

1. Go to Customization > Plug-ins > Plug-in Implementations > New.
2. Create a JavaScript file following the sample script.
3. Select the JavaScript file you created, from the **Script File** field and click **Create Plug-in Implementation** button.
4. Select the Outbound Validation Plug-in type from the Select Plug-in Type record.
5. Type the information in the required fields on the Plug-in Implementation record.

6. Click **Save**.

an Outbound Validation plug-in implementation record is created, you can select it in the E-Document Template record. For more information, see the help topic Custom Plug-in Creation.

Creating a Custom Plug-in Implementation for E-Document Custom Data Source

A custom plug-in implementation for custom data source will enable you to add custom data sources to an e-document template. With a custom data source plug-in specified in an e-document template, you can add to e-documents more field values from the transactions that the e-documents will be generated from.

You must create a custom plug-in implementation for custom data source first and then implement it in NetSuite so that it will be available for selection on the e-document template record. Create a JavaScript file for the custom data source plug-in implementation. The JavaScript file must be compatible with SuiteScript 2.0.

Note: This sample script uses the `define` function, which is required for an entry point script (a script you attach to a script record and deploy). You must use the `require` function if you want to copy the script into the SuiteScript Debugger and test it. For more information, see the help topic SuiteScript Debugger SuiteScript Debugger.

The following code is a sample custom plug-in implementation for e-document custom data source.

```
* @NApiVersion 2.x
 * @NScriptType plugintypeimpl
 * @NModuleScope Public
 */
define(["N/render"], function(nsrender) {
  /**
  * inject - This function will provide the custom data source during the generation process
  * @param {Object} params
  * @param {String} params.transactionId
  * @param {Object} params.transactionRecord
  * @param {Number} params.userId
  *
  * @returns {Object} result
  * @returns {render.DataSource} result.alias
  * @returns {string} result.format
  * @returns {Object | Document | string} result.data
  */

  function inject(params) {
    var txnRecord = params.transactionRecord;
    var txnId = params.transactionId;
    var userId = params.userId
    var customObj = {};
    log.debug("Custom Object", customObj);
    return {
      customDataSources: [
        {
          format: nsrender.DataSource.OBJECT,
          alias: "custom",
          data: customObj
        }
      ],
    };
  }

  return {
    inject: inject
  };
});
```

This script takes the input parameters from the JSON object. The parameters of this object are listed in the following table.

Parameter	Type	Description	Remarks
transactionRecord.type	String	The type of transaction being validated.	The id and type parameters provide information about thetransaction being validated. You can use themtogether to reference any field value of thetransaction using search.lookupFields(options) or record.load(options)
transactionRecord.id	Number	The internal ID of the transaction being validated.	
transactionId	String	The internal ID of the transaction being validated.	The id provide information about the transaction's internal ID.
userId	Number	This field holds the internal ID of the current logged in user.	This field value can be used wherever there is a requirement to refer to the current logged in user. For example, to update e-document audit trail by shared module API, userId can be used in owner property.

Important: The custom data source plug-in implementation script must have the `@NSScriptType plugintypeimpl`.

After creating the script for plug-in implementation, upload it to Customization > Plug-ins > Plug-in Implementations > New. The type of the custom plug-in implementation must "Custom Data Source for E-Document". For more information, see the help topics Custom Plug-in Creation, TemplateRenderer.addCustomDataSource(options) and Using Custom Data Sources for Advanced Printing.

The following are guidelines for custom data sources:

- Naming convention - Element names must not begin with digits
- Do not add a large amount of data to the data object in customDataSource. This may result in an Out of Memory error.

If the datasource provided by your implementation is in the following format:

```
return {
      customDataSources: [
        {
          format: nsrender.DataSource.OBJECT,
          alias: "custom",
          data: {isOneWorldEnabled: true}
        }
      ],
    };
```

You can now include the following custom data in the e-document template by using the datasource format.

- For XML e-document:

```
<xml> ${custom.isOneWorldEnabled} </xml>
```

- For JSON e-document:

```
{
"key": "${custom.isOneWorldEnabled}"
}
```

QR String Generation

The QR string generation logic can be implemented in Custom Data Source Plug-in Implementation. After successful implementation of the QR logic, generated string can be populated in **custbody_qr_string**. It is hidden and present in transaction record, using Custom Data Source as illustrated in Sample QR Code Plug-in Implementation(Custom Data Source). This populated field can be used in Advanced PDF/HTML Template to display QR code in generated e-document PDF.

QR Code Preview

To generate a preview of QR code in the transaction record under **E-Document** subtab, **custbody_qr_code** field (rich text) can be updated with img tag containing the source of data URL for the QR code value as illustrated in Sample QR Code Plug-in Implementation(Custom Data Source). The data URL for QR code value can be generated using any library which is capable of generating data URL from a string.

Sample QR Code Plug-in Implementation(Custom Data Source)

This is an example of QR Code Plug-in implementation (CDS). It shows example usage where SS2.0 can be used to:

- Show QR code preview in Transaction form by generating data URL and using it to update **custbody_qr_code** field. This is a rich text field that can be used to display the QR code preview on transaction record.
- Populate **custbody_qr_string**, a hidden field that can be used as a data source in any Advanced PDF/HTML Template to display QR Code in generated PDF.

```
/**
 * @NApiVersion 2.x
 * @NScriptType plugintypeimpl
 * @NModuleScope Public
 */
// Give the relative path of qrcode-generator module responsible for generation of data URL from QR code value
define(["N/encode", "N/render", "N/record", "./qrcode-generator"], function (encode, nsrender, record, qrCodeGen) {

    // Implement the logic to return QR string from the input QR data passed
    function getQRCodeString(inputQRData) {
        var qrString = '';
        /*
        * QR logic implementation goes here, with 'inputQRData' parameter object containing the required properties for generating QR string stored in qrString variable
        */
        return qrString;
    }

    // Function returns the encoded base 64 value of the string passed
    function getBase64(qrstring) {
        return encode.convert({
            string: qrstring,
            inputEncoding: encode.Encoding.UTF_8,
            outputEncoding: encode.Encoding.BASE_64_URL_SAFE
        });
    }

    // Function returns img element which contains the QR Code preview image
    function getQRCodePreview(qrBase64) {
        var qrDataUrl = '';
        /*
        * Here 3rd party library can be used to convert the 'qrBase64' string to data url generation from QR code value, as 'qrCodeGen' is imported in this example
        */

        if (qrDataUrl) {
            return '<img alt="QR code" src="'+ qrDataUrl + '" style="display: inline-block; width: 80px; height: 80px;"></img>';
        }
```

```
        return '';
    }

    function inject(params) {
        var tranType = params.transactionRecord.type;
        var tranId = params.transactionRecord.id;

        // Hidden QR string field in transaction record
        var QR_CODE_FLD = "custbody_qr_string";
        // QR preview field in transaction record
        var QR_CODE_PREVIEW_FLD = "custbody_qr_code";
        var updateQRFlds = {};
        var qrCustomObj = {};

        /*
        * Read the fields of transaction required for QR value generation and pass them to getQRCodeString function as shown below where QR logic can be implemented
        * Ex:
        * var data = {
        *     sellerName: subsidiaryLegal,
        *     vatNumber: vatRegistrationNo.toString(),
        *     invoiceTimeStamp: transDate.toISOString(),
        *     invoiceTotal: totalAmount.toString(),
        *     invoiceVatTotal: totalTax.toString(),
        * };
        */

        //Pass the QR Code input data to return QR string
        var qrString = getQRCodeString(data);

        //Pass the QR string to get the encoded Base 64
        var qrBase64 = getBase64(qrString);

        if(qrBase64)
        {
           // qrCustomObj will be sent as part of inject function response which can be used in e-document template which is shown in the below example
            qrCustomObj = {
                qrCode : qrBase64
            };

            // Update QR hidden text field to display in generated e document PDF
            updateQRFlds[QR_CODE_FLD] = qrBase64;

            // Update QR preview field only to display QR code in transaction form under E-document subtab
            updateQRFlds[QR_CODE_PREVIEW_FLD] = getQRCodePreview(qrBase64);

            // Submitting the QR fields which needs to be updated
            record.submitFields({
                type: tranType,
                id: tranId,
                values: updateQRFlds
            });
        }

        return {
            customDataSources: [
                {
                    format: nsrender.DataSource.OBJECT,
                    alias: "qrData",
                    data: qrCustomObj
                }
            ],
        };
    }

    return {
        inject: inject
    }
})
```

You can now include the following custom data of QR code in the e-document template by using the data source format with respect to the alias provided.

- For XML e-document:

```
<xml> ${qrData.qrCode} </xml>
```

- For JSON e-document:

```
{
"key": "${qrData.qrCode}"
}
```

Editing E-Document Templates

To edit e-document templates, go to Setup > E-Documents > E-Document Templates. Open the e-document template in edit mode and modify information as needed, then click **Save**.

Note: The **Transaction Type** field in the e-document template becomes unavailable if the e-document template has already been assigned to one or more transaction records. This field remains unavailable unless you remove the e-document template from all transactions.

Note: You cannot edit the native or default templates locked by Localization SuiteApps. In such cases, you can create a copy to modify and use the newly created templates.

E-Document Certification in the Outbound Process

Most countries that implement electronic invoicing in e-commerce require that businesses or companies submit their e-documents to a certification authority or tax agency. Certification authorities check the validity of e-documents and their compliance to any prescribed format and content. Certified e-documents are returned to the sending company for use in any other business requirements.

The outbound component of the Electronic Invoicing SuiteApp supports the sending of e-documents to authorities, organizations, or a tax agency that provide certification services.

Before e-documents can be sent for certification, the administrator must first assign a certification sending method by checking the **Sending Method for Certification** box on a sending method record. The certification sending method is a different sending method from the sending method applied at the transaction level. The certification sending method is applied to the transaction types and subsidiaries selected on the sending method record. Whether the certification sending method will use email or web service channel, the administrator must specify the certification services provider or authority as the recipient. For more information, see Creating an E-Document Sending Method Record.

Note: Only one certification sending method must be associated with a combination of subsidiaries and transactions you select.

E-documents generated from transaction types and subsidiaries with an assigned certification sending method will display the Certify E-Document button on the record. Clicking the Certify E-Document button sends the e-document to the certification authority.

On the other hand, e-documents generated from transaction types and subsidiaries without an associated certification sending method will display the Send E-Document button.

NetSuite receives successfully certified e-documents, of which certified XML or JSON file you can view from the E-Document subtab. The E-Document Status field on the E-Document subtab indicates the result of e-document certification. If certification fails, details of errors are logged on the E-document Audit Trail.

Note: If you have created your own certification sending method, you must store the certified copy of the XML or JSON in the File Cabinet and update the file ID of the certified e-document in the transaction body field certified e-document.

Certified e-documents can be sent to customers, vendors or any third party as recipients using the standard sending method assigned to the transaction. With the appropriate sending method selected, you can click the Send E-Document button on the transaction. For more information, see Sending the E-Document of a Single Transaction.

E-Document Network Status Overview

The Electronic Invoicing SuiteApp enables you to get e-documents certified from a regulatory body through an API. It is usually generated from a transaction. You can also get network status updates and display the relevant details under the **E-Document** subtab in the transactions.

The following table describes the transaction body fields which contain the network information.

Network Field	Field ID	Description
Network Reference Id	custbody_ei_network_id	It is a unique number generated by network API while certifying or sending e-document.
Network Name	custbody_ei_network_name	It is the name of a network. For instance, PEPPOL, SAT, ARIBA.
Network Status	custbody_ei_network_status	It displays a keyword representing one of the possible network statuses used by the network.
Network Status Updated On	custbody_ei_network_updated_date_time	It displays the time stamp (date and time) of the updated status.

An administrator must implement the getStatus method in the sending method to invoke e-document network status feature. For more info, read Creating E-Document Sending Methods

To use this feature for custom transaction types (CTT), you must register those custom transaction types again.

- To handle the rejected network status for a transaction, the E-Document Status field (custbody_psg_ei_status) of transaction is set to Certification Data Error. It occurs when the field has Rejected status. If certification is enabled, then set the status to Sending Failed.
- Transfer Order transaction type will not be supported to get e-document network status.

Providing Access to Get Network Status for Custom Roles

The **Get Network Status** button is hidden by default for custom roles.

To enable it, the administrator can perform the following steps:

1. Go to Setup > Users/Roles > Manage Roles.
2. Click the **Customize** or **Edit** hyperlink of the role to which you want to give get network status permission.

3. Under the **Authentication** subtab, check the **Allow Manual Updation of Network Status box**.
4. Click **Save**.

Creating E-Document Sending Methods

An administrator must create e-document sending methods so that they can be available for selection on the transaction record and the e-document package record. Sending methods must be implemented as custom plug-in implementations instead of scripts.

Before users can send e-documents, an administrator must first assign sending methods to the e-document package record. This will enable users to select an e-document sending method on the **E-Document** subtab of the customer's transaction records.

An administrator can select an employee whose name and email address should appear as the sender of e-documents sent by your company or subsidiary. Selecting a designated e-document sender is optional. If there is no designated sender, the system uses the name and email address of the user who sent the e-document as the sender. See Selecting a Designated E-Document Sender.

Another type of sending method, the certification sending method, supports the sending of e-documents to certification authorities, which certifies e-documents for validity and compliance to business requirements. For more information, E-Document Certification in the Outbound Process.

To create sending methods, see the following topics:

- Setting Up an Email Sending Method for E-Documents
- Creating Custom Methods for Sending E-Documents
  - Creating a Script for Sending E-Documents
  - Creating an E-Document Sending Method Record
  - Editing an E-Document Sending Method Record

Setting Up an Email Sending Method for E-Documents

Note: The system can send an e-document by email to a maximum of 10 recipients for each customer or vendor. The system counts each contact added as a recipient. If you add the same contact multiple times, each instance is considered one recipient.

You can send e-documents by email using the **NetSuite Email Custom Plugin** sending method included in the Electronic Invoicing SuiteApp. This sending method is automatically associated with the default e-document package record also provided by the SuiteApp, and cannot be associated with any other e-document package. Also, the **NetSuite Email Custom Plugin** sending method cannot be edited or deleted.

Before you can send e-documents by email to a customer or vendor, the customer or vendor record must be assigned an e-document package that has an email sending channel. You must also define the email recipients for your customer or vendor. For information, see Creating E-Document Packages and Defining E-Document Email Recipients.

After you send e-documents by email, the system sends you a notification informing you that the e-document sending process is finished. If there are errors found, an error report is included in the notification.

The **E-Document Audit Trail** subtab under the **E-Document** subtab of the transaction shows one of the following statuses to indicate whether the e-document was sent successfully or not:

- **Sent** – This status means the e-document was successfully sent. The **Details** column shows the email addresses of the sender and recipients.
- **Sending Failed** – This status means the e-document was not sent. Information about sending errors are shown in the **Details** column. You must fix the errors before you can successfully resend the e-document.

  For more information, see Outbound E-Document Sending Errors.

Creating Custom Methods for Sending E-Documents

An administrator can create various custom methods for sending e-documents to different customers and vendors. You can use the custom sending methods to get status updates from a network.

To create a custom method for sending e-documents, the administrator must first create an e-document sending method plug-in implementation and then create an e-document sending method record for that implementation.

After custom sending methods are created, they become available for selection on e-document package records and transaction records.

See the following topics:

- Creating a Custom Plug-in Implementation for Sending E-Documents
- Creating a Script for Sending E-Documents
- Creating an E-Document Sending Method Record
- Editing an E-Document Sending Method Record

Creating a Custom Plug-in Implementation for Sending E-Documents

A custom plug-in implementation for sending e-documents must be created so that it will be available for selection on the sending method record.

Create a JavaScript file for the custom plug-in implementation. The JavaScript file must be compatible with SuiteScript 2.0.

Note: This sample script uses the `define` function, which is required for an entry point script (a script you attach to a script record and deploy). You must use the `require` function if you want to copy the script into the SuiteScript Debugger and test it. For more information, see the help topic SuiteScript Debugger.

The following code is a sample custom plug-in implementation for sending e-documents.

```
/**
* send - This function is the entry point of our plugin script
* @param {Object} plugInContext
* @param {String} plugInContext.scriptId
* @param {String} plugInContext.sendMethodId
* @param {String} plugInContext.eInvoiceContent
* @param {Array}  plugInContext.attachmentFileIds
* @param {String} plugInContext.customPluginImpId
* @param {Number} plugInContext.batchOwner
* @param {Object} plugInContext.customer
* @param {String} plugInContext.customer.id
* @param {Array}  plugInContext.customer.recipients
* @param {Object} plugInContext.transaction
* @param {String} plugInContext.transaction.number
* @param {String} plugInContext.transaction.id
* @param {String} plugInContext.transaction.poNum
* @param {String} plugInContext.transaction.tranType
```

```
* @param {Number} plugInContext.transaction.subsidiary
* @param {Object} plugInContext.sender
* @param {String} plugInContext.sender.id
* @param {String} plugInContext.sender.name
* @param {String} plugInContext.sender.email
* @param {Number} plugInContext.userId
*
* @returns {Object}  result
* @returns {Boolean} result.success
* @returns {String}  result.message
*/
        function send(pluginContext) {

            var MSG_NO_EMAIL = translator.getString("ei.sending.sendernoemail");
            var MSG_SENT_DETAILS = translator.getString("ei.sending.sentdetails");

            var senderDetails = pluginContext.sender;
            var customer = pluginContext.customer;
            var transaction = pluginContext.transaction;
            var recipientList = customer.recipients;
            var result = {};
            var parameters;
            if (!senderDetails.email) {
                parameters = {
                    EMPLOYEENAME: senderDetails.name
                };
                stringFormatter.setString(MSG_NO_EMAIL);
                stringFormatter.replaceParameters(parameters);
                result = {
                    success: false,
                    message: stringFormatter.toString()
                };
            } else {
                var invoiceSendDetails = {
                    number: transaction.number,
                    poNumber: transaction.poNum,
                    transactionType : transaction.type,
                    eInvoiceContent: pluginContext.eInvoiceContent,
                    attachmentFileIds: pluginContext.attachmentFileIds
                };
                notifier.notifyRecipient(senderDetails.id, recipientList, invoiceSendDetails);

                parameters = {
                    SENDER: senderDetails.email,
                    RECIPIENTS: recipientList.join(", ")
                };
                stringFormatter.setString(MSG_SENT_DETAILS);
                stringFormatter.replaceParameters(parameters);

                result = {
                    success: true,
                    message: stringFormatter.toString()
                };

            }

            return result;

        }

        return {
            send: send
        };
    });
```

Important: The sending method custom plug-in script must have the `@NSScriptType plugintypeimpl`.

After creating the script for plug-in implementation, upload it to Customization > Plug-ins > Plug-in Implementations > New. The type of the custom plug-in implementation must be “Sending Plugin”. For more information, see the help topic Custom Plug-in Creation.

Creating a Script for Sending E-Documents

Important: Sending methods must be created as custom plug-in implementations instead of scripts. You must recreate existing sending method scripts as new custom plug-in implementations of the type 'Sending Plugin'. For more information, see Creating a Custom Plug-in Implementation for Sending E-Documents. The system will not support sending method scripts in NetSuite 2019.2.

An e-document sending method script must be a JavaScript file that is compatible with SuiteScript 2.0.

The script must return an object with the following function:

send(scriptContext)

Description	Executed when sending an e-document.
Returns	A result object.

Parameters

Note: The `scriptContext` and `result` parameters are JavaScript objects.

Parameter	Type	Required / Optional	Description
`scriptContext.scriptId`	string	required	The ID of the document in the file cabinet
`scriptContext.sendMethodId`	string	required	The ID of the customer's or vendor's selected sending method
`scriptContext.eInvoiceContent`	string	required	The e-document content as a string **Note:** This content is the generated e-document.
`scriptContext.attachmentFileIds`	array of strings	optional	The internal ID of the generated e document pdf file in file cabinet
`scriptContext.customPluginImpId`	string	required	The id (Field ID: scriptid) of the custom plug-in implementation of type sending plug-in set for the transaction.
`scriptContext.transaction.id`	string	required	The ID of the e-document transaction **Note:** The transaction id specified here is the ID of the document from which the e-document was generated.
`scriptContext.transaction.number`	string	required	The document number of the e-document transaction
`scriptContext.transaction.poNum`	string	optional	The PO/check number of the e-document transaction
`scriptContext.sender.id`	string	required	The ID of the designated sender of the e-document
`scriptContext.sender.name`	string	required	The name of the designated sender of the e-document

Parameter	Type	Required / Optional	Description
scriptContext.sender.email	string	required	The email address of the designated sender of the e-document
scriptContext.userId	number	required	This internal ID of the current logged in user, it can used wherever there is a requirement to refer to the current logged in user. For example, to update e-document audit trail by shared module API, userId can be used in owner property.

Note: This sample script uses the `define` function, which is required for an entry point script (a script you attach to a script record and deploy). You must use the `require` function if you want to copy the script into the SuiteScript Debugger and test it. For more information, see the help topic SuiteScript Debugger.

The following code is a sample script for sending e-documents.

```
/**
 * @NApiVersion 2.x
 * @NModuleScope Public
 */
define(["N/record"], function(record, error) {
    return {
        /**
     * send - Sample implementation: This will copy the e-document content to the document's
     Memo field
     *
     * @param {Object} plugInContext
     * @param {String} plugInContext.scriptId
     * @param {String} plugInContext.sendMethodId
     * @param {String} plugInContext.eInvoiceContent
     *
     * @param {Object} plugInContext.customer
     * @param {String} plugInContext.customer.id
     * @param {String[]} plugInContext.customer.recipients
     *
     * @param {Object} plugInContext.transaction
     * @param {String} plugInContext.transaction.id
     * @param {String} plugInContext.transaction.number
     * @param {String} plugInContext.transaction.poNum
     *
     * @param {Object} plugInContext.sender
     * @param {String} plugInContext.sender.id
     * @param {String} plugInContext.sender.name
     * @param {String} plugInContext.sender.email
     *
     *
     * @returns {Object} result
     * @returns {Boolean} result.success: determines
     * @returns {String} result.message: a failure message
     */
        send: function(plugInContext) {
            var result = {
                success: true,
                message: "Success"
            };
            try {
                var rec = record.load({
                    type: record.Type.INVOICE,
                    id: plugInContext.transaction.id,
                });
                rec.setValue({
                    fieldId: "memo",
                    value: [
                        "Script ID: " + plugInContext.scriptId,
```

```
                        "Customer: " + plugInContext.customer.name,
                        "Transaction: " + plugInContext.transaction.number,
                        "Sender: " + plugInContext.sender.name,
                        "Recipients: " + plugInContext.customer.recipients.join("\n"),
                        "Content: " + plugInContext.eInvoiceContent].join("\n\n")
                });
                rec.save();
            } catch (e) {
                result.success = false;
                result.message = "Failure";
            }
            return result;
        }
    };
});
```

The sending method gets network status updates and returns document's unique identifier as soon as the document is being sent to the network. The method must also return an additional key networkStatus in the send method implementation response.

```
networkStatus:{
              referenceId:"e34fedf343ferterjty",
              name:"SAT",
          status: "Certification Processed",
              updateDateTime: new Date("30/11/2022 3:32:14 PM")
  }
```

These are the details of parameters added in the response:

Parameter	Type	Description
networkStatus.referenceId	String	It is a unique number generated by network API while certifying or sending e-document.
networkStatus.name	String	It is the name of a network. For instance, PEPPOL, SAT, ARIBA.
networkStatus.status	String	It displays a keyword representing one of the possible network statuses used by the network.
networkStatus. updateDateTime	String	It displays the time stamp (date and time) of the updated status.

The following code is a sample script of a sending method which returns a network status object.

```
{
             success: true,
             message:"Sent E-Document successfully",
             networkStatus:{
               referenceId:"pl335ead",
               name:"PEPPOL",
               status: "Processed",
               updateDateTime: new Date("30/11/2022 3:32:14 PM")
             },
};
```

The script returns an object with the function getStatus when the sending method is used to get status updates from an API.

getStatus(scriptContext)

Description	Executed to get the network status updates.
Returns	A result object.

Parameters

Parameter	Type	Required / Optional	Description
`scriptContext.transaction.id`	string	required	The ID of the e-document transaction **Note:** The transaction id specified here is the ID of the document from which the e-document was generated.
`scriptContext.transaction.type`	string	required	The type of the e-document transaction.

getStatus method: Response Parameters

Parameter	Type	Required/Optional	Description
success	boolean	required	In case of successful communication with network status check API, it is set as true. In all failed communication cases, it is set as false.
message	string	required	Message displayed in the **E-document Audit Trail**.
networkStatus.status	string	required if success parameter is true	Keyword representing one of the possible network statuses used by the network.
networkStatus.updateDateTime	string	required if success parameter is true	Time stamp (date and time) for the status

The following code is a sample script of returned response for getStatus function.

```
{
            success: true,
            message: "Get the latest network status of e-document is successful",
            networkStatus:{
              status: "Processed",
              updateDateTime: new Date("30/11/2022 3:32:14 PM")
            },
}
```

Important: To prevent permission errors, make sure the sending method script has the **@NModuleScope Public** JSDoc .

Creating an E-Document Sending Method Record

Make sure that you have created a sending method custom plug-in implementation before you create e-document sending method records. For more information, see Creating a Custom Plug-in Implementation for Sending E-Documents.

To create an e-document sending method record:

1. Go to Setup > E-Documents > E-Document Sending Methods > New.

2. In the **Name** field, enter a name for the e-document sending method.
3. In the **E-Document Package** field, select the e-document package you want to associate this sending method with. For more information, see Creating E-Document Packages.
4. In the **E-Document Sending Method Plugin Implementation** field, select the e-document plug-in implementation for this method.
5. In the **Sending Channel** field, enter the sending channel to use for this method. For example, **email**, **SOAP** or **REST**.

   Note: If the sending channel is **email** (case-sensitive), the system validates the email recipients upon saving the customer or vendor record and when sending the e-document.
6. In the **Transaction Type** field, select one or more transaction types for which this sending method will be used. To select multiple transaction types, press and hold the **Ctrl** key while selecting the transaction types.

   The Transaction Type field only displays the transaction types applicable to or supported by outbound e-document sending, which include:

   - Cash Refund
   - Cash Sale
   - Credit Memo
   - Customer Payment
   - Estimate
   - Invoice
   - Item Fulfillment
   - Purchase Order
   - Return Authorization
   - Registered Custom Transaction Types
   - Transfer Order
   - Vendor Credit or Bill Credit

   For more information, see Transactions and Processes Supported by the Electronic Invoicing SuiteApp.

   Note: The selected transaction types cannot be modified after the sending method has been used in a transaction. You must remove the e-document sending method from the transaction before you can modify this field.
7. In the **Subsidiary** field, select the subsidiaries that this sending method will be associated with. To select multiple subsidiaries, press and hold the **Ctrl** key while selecting the subsidiaries.

   If only this sending method is associated with a subsidiary, the supported transactions of that subsidiary will display this sending method on the E-Document Template field on the E-Document subtab. For more information, see Multi-subsidiary Support in the Outbound Process.
8. (Optional) If the sending method is to be used for e-document certification, check the **Sending Method for Certification** box. For more information, see E-Document Certification in the Outbound Process.

   Note: Only one certification sending method must be assigned to a combination of subsidiaries and transactions you select.
9. Click **Save**.

This sending method can now be selected on e-document package records.

If the **Inactive** box is checked, this record will not be available for selection.

Editing an E-Document Sending Method Record

To edit an e-document sending method record, go to Setup > E-Documents > E-Document Sending Methods. Open the e-document sending method in edit mode and modify information as needed, then click **Save**.

Sending method records must reference sending method plug-ins, instead of scripts, from the **E-Document Sending Method Plugin Implementation** field. Existing sending method scripts must be recreated as new custom plug-in implementations of the type 'Sending Plugin'. For more information, see Creating a Custom Plug-in Implementation for Sending E-Documents.

Important: The system will not support sending method scripts in NetSuite 2019.2, but until that time, you can still edit and use existing sending method scripts.

Note: The **Transaction Type** field in the e-document sending method is disabled if the e-document sending method has been assigned to one or more transaction records. To enable the field, you must remove the e-document sending method from all transactions.

E-Document Email Custom Templates

An administrator can create custom templates to specify default subject and body text for e-document emails. Users can use these templates when sending e-documents by email to any entity selected in outbound e-document sending process.

After the administrator creates the e-document email templates, you can select them from the **E-Document Email Customization** dropdown list. You can find the dropdown list on the Subsidiary record or Company Information page in OneWorld accounts and on Company Information page in Single Instance accounts.

For more information, see Creating an E-Document Email Custom Template and Selecting an E-Document Email Custom Template.

Creating an E-Document Email Custom Template

An administrator can create one or more custom templates in the E-Document Email Customization Template page.

To create an e-document email custom template:

1. Go to Setup > E-Documents > E-Document Email Customization> New.
2. In the **Name** field, enter a name for the e-document email template.
3. In the **Email Subject Text** field, enter text for the subject of the email. You can also enter additional information using the following placeholders as needed:
   - {TRANTYPE} — Transaction type

- {TRANID} — Transaction ID
- {PONUM} — Transaction purchase order number
- {COMPANYNAME} — Company name

> **Note:** You must enter the placeholders in capital letters along with the braces {}. There should be no spaces inside the braces.

4. In the **Email Body Text** field, enter text for the body of the email. You can also enter additional information using the following placeholders as needed:
   - {TRANTYPE} — Transaction type
   - {TRANID} — Transaction ID
   - {PONUM} — Transaction purchase order number
   - {COMPANYNAME} — Company name

   > **Note:** You must enter the placeholders in capital letters along with the braces {}. There should be no spaces inside the braces.

5. Click **Save**.

   The template will be available for selection on the Subsidiary record or Company Information page. You can use the template only if you select **Default** option in the **E-Document Sending Method Plugin Implementation** field on the E-Document Sending Method record.

> **Note:** If you do not create a template in the E-document Email Customization record, then a system generated default template is used in the e-document email. This email is sent to customers or vendors during the outbound e-document sending process.

Selecting an E-Document Email Custom Template

After creating the custom templates for e-document email, an administrator can select the templates on the Subsidiary record or Company Information page. The selected template will be used in sending the e-documents by email to customers or vendors.

To select an e-document email custom template on the Subsidiary record:

1. Go to Setup > Company > Subsidiaries.
2. Click **Edit** next to the required subsidiary.
3. From the **E-document Email Customization** dropdown list, select the required custom template.
4. Click **Save**.

To select an e-document email custom template on the Company Information page:

1. Go to Setup > Company > Company Information.
2. From the **E-document Email Customization** dropdown list, select the required custom template.
3. Click **Save**.

Note: E-document email custom templates can be selected on both Subsidiary record and Company Information page for the following combinations:

- If you select a different template on both Subsidiary record and Company Information page, then the template you selected on the Subsidiary record is used.
- If you select a template on the Company Information page and not on the Subsidiary record, then the same template is applied on transactions of all the subsidiaries.
- If you select a template on the Subsidiary record for a transaction, then the selected template is used for all transactions that are created for this subsidiary. It overrides the template selected on the Company Information page.

Note: If you do not select a template either on Subsidiary record or Company Information page, then a system generated default template is used in the e-document email. This email is sent to customers or vendors during the outbound e-document sending process.

Selecting a Designated E-Document Sender

An administrator can select an employee whose name and email address should appear as the sender of e-documents sent by your company or subsidiary. This task is optional. If there is no designated sender, the system uses the name and email address of the user who sent the e-document as the sender.

To select a designated e-document sender:

1. Go to Setup > Company > Company Information.

   If you have a OneWorld account, go to Setup > Company > Classifications > Subsidiaries, and click the edit link to open a subsidiary record.
2. In the **E-Document Sender** field, select an employee.
3. Click **Save**.

Setting Up Custom Roles to Send E-Documents

The following standard roles can send and re-send e-documents in bulk:

- A/P Clerk
- A/R Clerk
- Accountant
- Administrator
- Bookkeeper
- CEO
- CFO
- Custom Accountant
- Retail Clerk

An administrator can give custom roles access to the bulk sending feature. The following conditions must be met to enable a custom role to send e-documents in bulk:

- The custom role has been added to the script audience for the **Outbound E-Invoicing Form SU** script.
- The custom role has Edit permissions for the transaction types for which your company generates and sends e-documents.
- The custom role has Edit permission for the Add E-Document Sending Batch custom record.

To grant custom roles access and permissions to send e-documents:

1. Add the custom roles to the script audience by performing the following steps:.
    a. Go to Customization > Scripting > Script Deployments.
    b. Click the Edit link of the script named **Outbound E-Invoicing Form SU**.
    c. On the **Audience** subtab, in the **Roles** field, select the custom roles that you want to give access to.
    d. Click **Save**.
2. Add the required permissions to custom roles by performing the following steps:
    a. Go to Setup > Users/Roles > Manage Roles.
    b. Click the Edit link of the custom role that you want to modify.
    c. Add permission to edit transaction types:
        i. On the **Permissions** subtab, go to the **Transactions** subtab.
        ii. In the **Permission** column, on a blank row, select a transaction type.
        iii. In the **Level** column, select **Edit**.
        iv. Click **Add** to add the row.
        v. Add other transaction types that the custom role must have access to.
    d. Add permission to edit the custom record used for bulk sending:
        i. On the **Permissions** subtab, go to the **Custom Record** subtab.
        ii. In the **Record** column, on a blank row, select **Add E-Document Sending Batch**.
        iii. In the **Level** column, select **Edit**.
        iv. In the **Restrict** column, select **Editing Only**.
        v. Click **Add** to add the row.
    e. Click **Save** to save the changes to the custom role.

Customizing Roles to Restrict E-Document Generation or Sending

By default, the Generate E-Document and Send E-Document buttons are visible on all outbound transaction types that E-Invoicing SuiteApp supports, regardless of the level of permission the user has on the transaction record.

The administrator can customize standard and custom roles to restrict their permission to manually generate and send outbound e-documents.

To restrict the permission of standard and custom roles to generate and send outbound e-documents:

1. Go to Setup > Users/Roles > User Management > Manage Roles.
2. Click the Customise or Edit link of the role that you want to restrict permission to generate or send outbound e-documents.
3. Clear the **Allow Manual Generation of E-Document** box or **Allow Manual Sending of E-Document** box.
4. If you want to allow this role to send e-documents for certification, check the **Allow Sending of E-Document for Certification** box; otherwise, clear the box.
5. Click **Save**.

   The Generate E-Document button or Send E-Document button will not be available on transaction records viewed by role. If you did not grant the role permission to send e-documents for certification, the Certify E-document button will not be available on transactions viewed by role.

Deploying the Bulk Generation Script for E-Documents

The Electronic Invoicing SuiteApp enables an administrator to deploy a script to generate multiple e-documents at a specific time, instead of generating e-documents for each transaction manually.

By default, the script's status is set to **Not Scheduled**. The administrator can set a schedule for the script to generate e-documents in bulk.

Note: Only an administrator can view, edit, and run the **Generate E-Document Content** script included in the SuiteApp.

The script generates e-documents for transactions that have the following e-document statuses:

- **For Generation** – The script generates e-documents for transaction records that have e-document templates.
- **Generation Failed** – The script regenerates e-documents for transaction records with assigned e-document templates that previously encountered errors during generation. Errors must be fixed before e-documents can be regenerated. For more information, see Outbound E-Document Generation Errors.

After the system generates e-documents, it updates the value in the **E-Document Status** field on the **E-Document** subtab of each transaction record. The field shows one of the following values:

- **Ready for Sending** – This status means the e-document was generated successfully and can be sent to the customer or vendor.
- **Generation Failed** – This status means the e-document was not generated. Information about generation errors are shown in the **Details** column of the **E-Document Audit Trail** subtab. The user must first fix the errors before the e-document can be regenerated for the transaction record.

Note: If e-document generation fails, NetSuite does not send an email notification to the administrator. Instead, it sends an email notification containing the error details to the user who created the transaction record. The user must fix the errors before an e-document can be regenerated for the transaction record.

For more information about generation errors, see Outbound E-Document Generation Errors.

See also Generating and Regenerating E-Documents in Bulk.

To deploy the bulk generation script for e-documents:

1. Go to Customization > Scripting > Script Deployments.
2. Click the Edit link of the **Generate E-Document Content MR** script.
3. Clear the **Deployed** box if you do not want to deploy the script yet. A script will not run in NetSuite until the **Deployed** box is selected.
4. Select a status in the **Status** field:
   - Testing
   - Not Scheduled
   - Scheduled

   For more information, see the help topic Setting Script Deployment Status.
5. Choose an event type on the **Schedule** subtab:
   - **Single event** – The script generates e-documents only one time.
   - **Daily event** – Enter the interval between days if this event is every day or every few days, or select every weekday if this event is every day except Saturdays and Sundays.

     Enter 1 as the interval if this event is every day, for example, or enter 2 if the event is every other day.
   - **Weekly event** – Enter the interval between weeks, and select the day of the week this event repeats on.
   - **Monthly Event** – If you want to generate e-documents on the same day of every month or every few months, enter the date the generation of e-documents repeats, and select the interval between months.

     If you want to generate e-documents on the same day of the week every month or every few months, select the week, the day of the week, and enter the interval between months.
   - **Yearly Event** – If you want to generate e-documents one time every year, select the month and day to generate e-documents, or select the week, day, and month.
   - **Start Date** – You must enter the date you want to schedule the generation of e-documents for.
   - **Start Time** – Enter the time you want the generation of e-documents to start.
   - **Repeat** – Select how often you want the script to generate e-documents. On the day the script is scheduled to run, e-documents are generated at the specified **Start Time** and repeat every *n* hours until midnight.
   - **End By** – Set the date the script stops generating e-documents.

     If you are scheduling a single event, e-documents will be generated on the date entered in the **Start Date** field.

     If you are scheduling a repeat event, e-documents will be generated according to the schedule you set starting on the date entered in the **Start Date** field and ending on the date entered in the **End By** field.
   - **No End Date** – Check this box if you want to repeat the schedule indefinitely.
6. Click **Save and Execute** to generate e-documents immediately. Click **Save** to generate e-documents according to the schedule you chose.

Deploying the Script for Scheduled Sending of E-Documents

Like the script for generating e-documents in bulk, the Electronic Invoicing SuiteApp also has a script for automatic scheduled sending of outbound e-documents. The script sends only outbound e-documents whose status is Ready for Sending.

Note: By default, only the administrator can view, edit, and run the Automatic Send E-Document MR script included in the SuiteApp, because the administrator has the permission to view, edit and run Script Deployment records.

By default, the script's status is set to Not Scheduled. The administrator can set a schedule for the script to send outbound e-documents.

To deploy the script for scheduled sending of e-documents:

1. Go to Customization > Scripting > Script Deployments
2. Click the Edit link of the **Automatic Send E-Document MR** script.
3. Clear the **Deployed** box if you do not want to deploy the script yet. A script will not run in NetSuite until the Deployed box is selected.
4. Select a status in the **Status** field:
   - Testing
   - Not Scheduled
   - Scheduled

   For more information, see the help topic Setting Script Deployment Status.
5. Choose an event type on the **Schedule** subtab:
   - **Single Event** – The script sends outbound e-documents only one time.
   - **Daily Event** – Enter the interval between days if this event should occur every day or every few days, or select every weekday if this event should occur every day except Saturdays and Sundays. For example, enter 1 as the interval if this event should occur every day, or enter 2 if the event should occur every other day.
   - **Weekly Event** – Enter the interval between weeks, and select the day of the week this event should be repeated.
   - **Monthly Event** – If you want to send e-documents on the same day of every month or every few months, enter the date when you want e-document sending to repeat and then select the interval between months. If you want to send e-documents on the same day of the week every month or every few months, select the week, the day of the week, and enter the interval between months.
   - **Yearly Event** – If you want to send e-documents one time every year, select the month and day, or select the week, day, and month.
   - **Start Date** – You must enter the date when you want to schedule the sending of e-documents .
   - **Start Time** – Enter the time when you want the sending of e-documents to start.
   - **Repeat** – Select how often you want the script to send e-documents. On the day the script is scheduled to run, e-documents will be sent at the specified Start Time and the process repeats every n hours until midnight.
   - **End By** – Set the date when the script should stop sending e-documents.

     If you are scheduling a single event, e-documents will be sent on the date entered in the Start Date field.

     If you are scheduling a repeat event, e-documents will be sent according to the schedule you set, starting from the date entered in the Start Date field and ending on the date entered in the End By field.
   - **No End Date** – Check this box if you want to repeat the schedule indefinitely.
6. Click **Save and Execute** to send e-documents immediately. Click **Save** to send e-documents according to the schedule you chose.

The script automatically searches for outbound e-documents whose status are Ready for Sending, and then sends them. After the outbound e-documents are sent, their status is changed to either Sent or Sending Failed. Details of any error during sending are recorded in the E-Document Audit Trail. Error notification is sent to the Recipient of E-Document Notifications. If Recipient of E-Document Notifications has not been defined yet, the notifications are sent to active administrators. The notification has a CSV file attached containing details of the error.

Scheduled Script for Getting Network Status of E-Documents

The Electronic Invoicing SuiteApp provides a script to automatically get network status for bulk transactions. The script has an additional parameter **Transactions Filtering Plugin ID** (id: custscript_ei_filtertransid). This parameter populates the implementation id of the Transactions Filtering Plugin ID field, which contains the logic for filtering transactions. By default, the script selects transactions if the following conditions are met:

1. The **Network Reference Id** must be populated.
2. The transactions must be created the current or previous day.

By default, the script's status is set to **Scheduled** and it runs every 2 hours. The administrator can set a different schedule as per their requirements.

Implementing Transaction Filtering Plug-in

The Electronic Invoicing SuiteApp has added a custom plug-in type called the Transaction Filtering Plugin (id: customscript_transactionfilterplugin). This plug-in contains the following method which the administrator must implement:

getFilteredTransactions: This method filters the transactions for which the **Automatic Get Network Status MR** script updates the statuses. The method returns an array of objects. The objects have two keys that contain the details of transactions.

The details of objects in the returned array are given in the following table:

Parameter	Type	Description
id	String	It is the unique number to identify the transaction.
recordType	String	It specifies the type of transaction.

In the default implementation, the **getFilteredTransactions** method returns the list of transactions where:

- The **Network Reference Id** is populated.
- The transactions are created the current or previous day.

```
/**
 * @NApiVersion 2.x
 * @NModuleScope Public
 * @NScriptType plugintypeimpl
 */
define(["N/search", "N/format"], function (search, format) {
  /**
```

```
 * getFilteredTransactions - This function is the entry point of our plugin script
 *
 * @returns {Array} transactions
 * @returns {Object} transaction
 * @returns {String} transaction.recordType
 * @returns {String} transaction.id
 */
function getFilteredTransactions() {
  var transactions = [];
  var yesterdayDate = new Date();
  yesterdayDate.setDate(yesterdayDate.getDate() - 1);
  var transSearch = search.create({
    type: search.Type.TRANSACTION,

    filters: [
      [
        "trandate",
        search.Operator.NOTBEFORE,
        format.format({
          value: yesterdayDate,
          type: format.Type.DATE,
        }),
      ],
      "and",
      ["custbody_ei_network_id", search.Operator.ISNOTEMPTY, ""],
      "and",
      ["mainline", "is", "T"],
    ],
  });
  var PAGE_SIZE = 1000;

  var searchResult = transSearch.runPaged({ pageSize: PAGE_SIZE });
  searchResult.pageRanges.forEach(function (pageRange) {
    var currPage = searchResult.fetch({ index: pageRange.index });

    currPage.data.forEach(function (result) {
      transactions.push({
        recordType: result.recordType,
        id: result.id,
      });
    });
  });
  return transactions;
}

return {
  getFilteredTransactions: getFilteredTransactions,
};
});
```

The administrator must add a plug-in implementation for the Transaction Filtering Plugin type. You must provide the id for implementation in the **Automatic Get Network Status MR** script's parameter **Transaction Filtering Plugin ID**.

Adding Transaction Filtering Plugin ID

The administrator must add a plug-in implementation for the Transaction Filtering Plugin type.

To add Transaction Filtering Plugin ID for bulk network status:

1. Go to Customization > Scripting > Script Deployments.
2. Click **Edit** on the **Automatic Get Network Status MR** script.
3. Go to **Parameters** subtab.
4. In the **Transactions Filtering Plugin ID** field, enter the relevant ID.
5. Click **Save**.

Updating E-Document Certification Statuses

E-document certification statuses are not automatically updated by the Electronic Invoicing SuiteApp. E-document certification statuses can be updated through the certification sending method implementation. If your company or organization is implementing its own e-document certification sending, you can use the format of the following sample script for capturing sending method return, which can be used to update the e-document certification statuses and log corresponding messages in the E-document Audit Trail.

Note: This sample script uses the `define` function, which is required for an entry point script (a script you attach to a script record and deploy). You must use the `require` function if you want to copy the script into the SuiteScript Debugger and test it. For more information, see the help topic SuiteScript Debugger.

The following code is a sample script to implement e-document certification status updates.

```
/**
* Copyright (c) 2017, Oracle and/or its affiliates.
*
* @NApiVersion 2.x
* @NScriptType plugintypeimpl
* @NModuleScope public
*/
define([], function() {
   /**
   * send - This function is the entry point of our plugin script
   * @param {Object} plugInContext
   * @param {String} plugInContext.scriptId
   * @param {String} plugInContext.sendMethodId
   * @param {String} plugInContext.eInvoiceContent
   * @param {Object} plugInContext.customer
   * @param {String} plugInContext.customer.id
   * @param {Array} plugInContext.customer.recipients
   * @param {Object} plugInContext.transaction
   * @param {String} plugInContext.transaction.number
   * @param {String} plugInContext.transaction.id
   * @param {String} plugInContext.transaction.poNum
   * @param {Object} plugInContext.sender
   * @param {String} plugInContext.sender.id
   * @param {String} plugInContext.sender.name
   * @param {String} plugInContext.sender.email
   * @param {Array} plugInContext.attachmentFileIds
   *
   * @returns {Object} result
   * @returns {Boolean} result.success
   * @returns {String} result.message
   */
   return {
       send: function(pluginContext) {
           var result = {
               success: true,
               message: '',
               eiStatus: {
                   "transactionId": plugInContext.transaction.id,
                   "transactionType": plugInContext.transaction.tranType,
                   "entity": customer.id,
                   "eDocStatus": "3",
                   "eventType": "3",
                   "details": "The e-Doc successfully certified and is ready for sending.",
                   "owner": plugInContext.sender.id,
                   "isUpdateFields": "true",
                   "extraFieldsForUpdate": {},
                   "bundleId": "",
                   "bundleName": ""
```

```
            }
        };
        return result;
    }
  }
});
```

If e-document certification is successful, the script will change the e-document status of the transaction to Ready for Sending and add the following details in the E-Document Audit Trail:

`The e-document was successfully certified and is ready for sending.`

The following table lists the parameters that the object eiStatus in the script takes up.

Parameter	Type	Description	Values
transactionId	String	ID of transaction record	Required
transactionType	String	Transaction type such as INVOICE, VENDOR_BILL, ITEM_FULFILLMENT, and others	Required
entity	String	ID of entity record; for example, Customer, Vendor, and others.	Required
eDocStatus	String	E-document Status Use numbers for status types. The following numbers can be used: ■ READY_FOR_SENDING : 3 ■ CERTIFICATION_IN_PROGRESS : 20 ■ CERTIFICATION_DATA_ERROR : 21 ■ CERTIFICATION_FAILED : 22	Required
eventType	String	E-document Audit Trail Event Use numbers for event types. The following numbers can be used: ■ READY_FOR_SENDING : 3 ■ CERTIFICATION_IN_PROGRESS : 20 ■ CERTIFICATION_DATA_ERROR : 21 ■ CERTIFICATION_FAILED : 22	Required
details	String	Details	Required
owner	String	ID of user or Employee	Required
isUpdateFields	String	Flag; if TRUE, the EI Status is updated, fields are provided in extraFieldsForUpdate field and an audit trail is created. Otherwise, an audit trail is created.	Required
extraFieldsForUpdate	JSON	A JSON object containing the key-value pair for the transaction body fields, where key is the field ID and `value` is the field value. It can be used to update any transaction body field required by a Country SuiteApp.	Required (can be an empty object)
bundleId	String	BundleId of the requester	Required
bundleName	String	BundleName of the requester	Required

The parameter details will be included in the E-Document Audit Trail.

Processing E-Documents Automatically for Individual Transactions

A user with required permissions can process e-documents automatically for individual transactions. All the e-document processes like generate, certify and send can be automated by clicking a single **Process E-Document** button. Before processing the e-documents automatically for a transaction, the transaction's e-document template, sending method and certification must be set up.

Tip: Processing e-documents automatically rather than manually can be useful in reducing the time and effort, especially when it is used along with auto-selection of template and sending method. For more information, see **Template and Sending Method Auto-selection** in Multi-subsidiary Support in the Outbound Process

The automatic processing of e-document feature is disabled by default. To enable this feature, an administrator must select an **E–Document Automation Type** from the dropdown available in the Automatic E-Invoicing tab on the E-Document Preferences page.

To process e-documents automatically for individual transactions:

1. Go to Setup > E-Documents > E-Document Preferences.
2. In the E-Document Preferences page, select one of the following options from the **E-Document Automation Type** dropdown present in the Automatic E-Invoicing tab for each subsidiary.
   - Generate, Certify, Send – The **Process E-Document** button becomes available on the transactions. It enables you to perform e-document's generate, certify and send processes by clicking the **Process E-Document** button.

     Note: E-document certification process is not performed if a valid certification sending method is not defined.
   - Generate, Certify – The **Process E-Document** button becomes available on the transactions. It enables you to perform e-document's generate and certify processes by clicking the **Process E-Document** button.
   - Certify, Send – The **Process E-Document** button becomes available on the transactions. It enables you to perform e-document's certify and send processes by clicking the **Process E-Document** button.
   - Disable – This option is selected by default. The **Process E-Document** button is not available on the transactions. The e-document's generate, certify and send processes must be performed manually using the **Generate E-Document**, **Certify E-Document** and **Send E-Document** buttons. For more information, see Overview of Outbound E-Document Process.

Only a user with administrator role an configure the e-document processes for automation. After this you can use the **Process E-Document** button on individual transactions to process the e-documents automatically for a subsidiary.

Electronic Invoicing Inbound Email Capture

Inbound electronic invoicing supports the capturing of vendor invoices in XML format sent through email. This means your vendors can send you invoices in XML through email, which you can directly view as inbound e-document records.

When the system receives email with an XML invoice attached, it first determines the vendor based on either the vendors' email domain or sender email address. If the vendor is determined, the system

automatically captures the email with the XML invoice and then stores the XML content as an inbound e-document record. Inbound e-document records from email capture are automatically added to the Inbound E-Documents List with the e-document status indicated.

If the vendor is not determined from the email e-document, or other errors are encountered, the system sends a notification to the Recipient of E-Document Notifications, informing that user of the problems encountered with the received email e-document. If the Recipient of E-Document Notifications is not set up, the notifications are sent to all active administrators.

Setting Up Inbound Email Capture

Set up the Inbound Email Capture feature in your account by performing the following tasks:

- Enable the Inbound Email Capture PIug-in.
- Set up Vendor E-Document Email Sender.

Enabling Inbound Email Capture Plug-in

The Electronic Invoicing SuiteApp comes with an inbound email capture plug-in that you must first enable.

To enable the inbound email capture plug-in:

1. Go to Customizations > Plug-ins > Manage Plug-ins.
2. On the Manage Plug-In Implementations page, check the **Inbound Email Capture PI** box.
3. Click **Save**.

Take note of the email address indicated, with the domain emails.netsuite.com. This is the email address where your vendors should send their XML invoices to. This email address is different for every account. It is that you set up an alias for this email address in your email facility, so that your vendors can easily record or remember the email address. For more information about setting up an email alias, see the help topic Create an Email Alias and Set Up Forwarding

To view details of events or errors associated with the inbound email capture plug-in, go to Customization > Plug-ins > Plug-in Implementations. Click the View link of the Inbound Email Capture PI and then on the Plug-In Implementation page, click the Execution Log subtab. For more information about the email capture plug-in, see the help topics Email Capture Plug-in Overview and Administering an Email Capture Plug-in Implementation

Aside from the plug-in execution log, the system also sends inbound e-document email capture notification messages to your designated recipient, informing that user of the details of any event or error associated with the inbound email capture plug-in.

Setting Up Vendor E-Document Email Sender

After enabling the e-document email capture plug-in, you must identify or assign the Vendor E-Document Sender in vendor records.

To set up the vendor e-document email sender

1. Edit a vendor record and click the **E-Document** subtab.
2. You can enter the email domain associated with the vendor in the **Sender Email Domain** field. If you enter an email domain in this field, the system will use the email domain to determine the vendor of the captured email e-documents.

> **Note:** The vendor email domain you enter must be unique for each vendor. No vendors must have the same email domain.

3. Enter the email address of the vendor's designated user in the **Vendor E-Document Email Sender** sublist. Add as many vendor sender email addresses as necessary. The system will use the email addresses you entered to determine the vendor of captured email e-documents.

> **Note:** Only designated vendor e-document email senders can create inbound e-documents.

4. If you want the system to use the email addresses you entered in **Vendor E-Document Email Sender** instead of the domain, for determining the vendor of captured e-document email, you can check the **Use Sender Email List** box. If you check the box, you must enter email addresses in the **Vendor E-Document Email Sender** field, which becomes required and replaces the domain as the basis for determining the vendor.
5. Click **Save**.

If you do not enter an email domain or check the Use Email Sender List box, the system will not be able to determine the vendor of the email e-document and an inbound e-document will not be created. In this case, the system will send a notification to the recipient you designated, informing that user that e-document email was received but no vendor was associated with the e-document.

To apply these settings to your existing vendor records, you can use CSV Import.

To update vendor records to apply or update vendor e-document email sender or recipient:

1. Go to Setup > Import/Export > Import Tasks > Import CSV Records.
2. In the **Import Type** field, select Custom Records.
3. In the **Record Type** field, select Vendor E-Document Email Sender or Recipient.
4. Click **Select**, and select the CSV file to upload.
5. Click **Next**.
6. On Import Options, select the appropriate Data Handling option. Click **Next**.
7. On Field Mapping, if you are updating the vendor e-document email sender, make sure that email and vendor are mapped to appropriate values. If you are updating the vendor e-document email recipient, make sure that the contact and vendor are mapped to appropriate values. Click **Next**.
8. In the **Import Map Name**, enter a unique name. Click **Save & Run**.

For more information about performing CSV Import, see the help topics CSV Imports Overview and Importing CSV Files with the Import Assistant.

Using SOAP Web Services for Inbound Processing

Inbound e-document processing supports SOAP web services, which offer advantages in interoperability, flexibility and security in sending information over the internet. By enabling the web service features in NetSuite, you can receive XML files sent through web service and then capture information in XML for subsequent creation of e-documents. You can set up vendor or employee records to authorize them as web service senders. As authorized web service senders, these entities can send single or multiple XML files in one web service request.

Setting Up Inbound E-Document SOAP Web Services

1. Go to Setup > Company > Setup Tasks > Enable Features.

a. On the **SuiteCloud** subtab, under SuiteTalk (Web Services), check **SOAP Web Services**.

b. Under Manage Authentication, check **Token-Based Authentication**.

c. On the **Web Presence** subtab, under Access, check **Vendor Access**.

d. Click **Save**.

For more information, see the help topics SuiteCloud Features and Commerce Features.

2. Create an integration record to generate authentication keys. Go to Setup > Integration > Integration Management > Manage Integrations > New. For more information, see the help topic Creating an Integration Record.

a. Enter or select values in the **Name** and **Description** fields. Retain the default value in the **State** field, which is **Enabled**.

b. On the **Authentication** subtab, check **Token-based Authentication**. The **User Credentials** box is checked by default, retain this.

c. Click **Save**.

The Consumer Key and Consumer Secret are generated and displayed on the integration record. Copy the Consumer Key and Consumer Secret to send to the vendors or parties that you want to authorize as web service senders.

3. If you upgraded the Electronic Invoicing SuiteApp from a previous version, make sure that the custom role, Inbound E-Document Web Service Role, is included in the target audience of the E-Document Inbound UE script deployment. But if the SuiteApp is a new installation, you do not have to perform these steps.

a. Go to Customization > Scripting > Scripts.

b. Click the Deployments link of the E-Document Inbound UE script and edit the record.

c. On the **Audience** subtab, under Roles, select **Inbound E-Document Web Service Role** to add it to the list of roles.

d. Click **Save**.

4. Assign the Inbound e-Document Web Services Role to the vendor or employee whom you want to authorize to be the sender of XML files through web service.

a. Edit the vendor record that you authorized as web service sender.

b. On the vendor record **E-Document** subtab, enter appropriate values in the **Web Service ID** and **Web Service Sender** fields.

For the Web Service ID, you can assign any unique identifier of the vendor, like the tax ID. For the Web Service Sender, you can choose from the list a vendor or an employee, whom a vendor designated to send e-documents of its behalf.

c. Click **Save**.

For more information, see the help topics Assigning Roles to an Employee and step 6 under the Access section in Creating a Vendor Record.

5. Create access tokens for each vendor or employee that you authorized as web service sender.

a. Go to Setup > Users/Roles > User Management > Access Tokens > New.

b. In the **Application Name** field, select the integration record that you created in step 2, which will be associated with the access token you will create.

c. In the **User** field, select the vendor or employee whom you assigned the Inbound e-Document Web Service Role.

d. In the **Role** field, select **Inbound e-Document Web Service Role**. The Token Name is automatically assigned a value.

e. Click **Save**.

The Token ID and Token Secret are generated and displayed on the Access Token page. Copy the values for the Token ID and Token Secret before going to another page.

> **Important:** For security reasons, the values for the Token ID and Token Secret are only displayed on the initial setup page. They cannot be retrieved from the system. If you did not take note or copied the Token ID and Token Secret values, you will need to create a new token.

For more information, see the help topic Getting Started with Token-based Authentication.

6. Send or convey the following web service connection and authentication details to authorized web service senders:
   - Consumer Key
   - Consumer Secret
   - Token ID
   - Token Secret
   - Account ID

     Go to Setup > Company Information. In the **Account ID** field, obtain the value that you need to send.
   - Inbound E-Document Web Service RESTlet External URL

     Go to Customization > Scripting > Scripts. View the Inbound E-Document Web Service RL, RESTlet. On the **Deployments** subtab, click **Inbound E-Document Web Service RL**. The External URL is displayed.
   - The vendor Web Service ID.

After setting up SOAP web services for inbound processing, you are ready to receive XML e-documents from SOAP web services. For more information, see Receiving E-Document XML Files from Web Service.

Web Service Sender Setup Tasks

The authorized web service sender can either be a vendor or an employee. Web service senders must also perform setup tasks before they can send e-document files through web service.

First, a web service sender must obtain the web service connection and authentication details from you. Then, they must meet the following requirements in constructing the web service request.

Requirements	Web Service Details
OAuth 1.0 Authentication	Use the authentication details from the NetSuite user: ■ Consumer Key ■ Consumer Secret ■ Token ID ■ Token Secret ■ Realm (Account ID)
OAuth Data	■ Consumer Key = generated from Netsuite ■ Signature Method = HMAC-SHA1 or HMAC-SHA256

Requirements	Web Service Details
	■ OAuth Nonce = unique string that is generated ■ Timestamp = timestamp in seconds ■ OAuth Version = 1.0 ■ OAuth Token = generated from Netsuite ■ OAuth Signature = Key is secret key, Base String. For more information, see the help topics Using TBA for RESTlet Authentication (OAuth) and Step One Obtain An Unauthorized Request Token.
Content-type header	Application/json
Request Method	POST

Web Service Request Body	JSON Object
	Sample code for sending a single XML file in one request:
	`{ "identifier": "vendor1", "fileName": "vendor1.xml", "content": "<XML>content</XML>" }`
	If multiple XML files must be sent in one web service request, the format must be an of JSON objects.
	Sample code for sending multiple XML files in one request:
	`[{ "identifier": "vendor1", "fileName": "vendor1-1.xml", "content": "<XML>content</XML>" }, { "identifier": "vendor1", "fileName": "vendor1-2.xml", "content": "<XML>content</XML>" }, { "identifier": "vendor2", "fileName": "vendor2.xml", "content": "<XML>content</XML>" }]`
	Important: There is a limit of 10MB per string used as RESTlet input or output. For more information, see the help topic RESTlet Governance.

Setting Up Custom Roles that can Convert Inbound E-Documents

An administrator can create custom roles or customize standard roles, to grant them the permission to perform conversion of inbound e-documents into transaction records.

To grant a role the permission to convert inbound e-documents into transaction records:

1. Set up the permission. Go to Setup > Users/Roles > User Management > Manage Roles. Select the role you want to grant permission to and click its Customize link. On the Role page, do the following:
   - On the **Custom Record** subtab, do the following:
     - In the Inbound E-Documents row, click the entry in the Level column. Select **Edit**.
     - In the E-Document Audit Trail row, click the entry in the Level column. Select **Edit**.
   - On the **Transactions** subtab, do the following:
     - In the Bills row, click the entry in the Level column. Select **Full**.
     - In the Purchase Order row, click the entry in the Level column. Select **View**.
   - On the **Lists** subtab, do the following:
     - In the Vendors row, click the entry in the Level column. Select **View**.
     - In the Perform Search row, click the entry in the Level column. Select **View**.
     - In the Documents and Files row, click the entry in the Level column. Select **View**.
   - Click **Save**.
2. Make the Convert button visible to the roles with permission to perform e-document conversion. To do so, the inbound e-document script deployment must be enabled. Go to Customization > Scripting > Script Deployments. On the Script Deployments page, do the following:
   - Click the Edit link of the script customdeploy_edoc_ue_inbound, **E-Document Inbound UE**.
   - On the **Audience** subtab, select the roles that the script will execute for. To select multiple roles, press and hold down the Ctrl key while selecting the roles.
   - Click **Save**.

     For more information about associating a role with script deployment, see the help topic Executing Scripts Using a Specific Role.
3. Assign the role with the permission to perform inbound conversion, to employees. See the help topic Assigning Roles to an Employee.

For more information, see the help topics Customizing or Creating NetSuite Roles and Setting Permissions for a Custom Record Type.

Inbound Validation Plug-ins

The Electronic Invoicing SuiteApp provides support for plug-ins that can perform inbound e-document validation according to the requirements of tax agencies or standards regulatory bodies.

You can create or customize a plug-in that communicates with external sites or systems of a third party, such as a tax agency, which validates e-documents and then returns the results to the plug-in. Or you can create or customize a plug-in that itself performs the required e-document validation.

An inbound e-document validation plug-in must be set up first and integrated with a vendor e-document package. With the validation plug-in set up, every time a user converts inbound e-documents to vendor bills, the plug-in automatically triggers the validation process that is transparent to the user. Depending on your implementation of the plug-in, it can either send the XML e-documents to a validating third party or validate them locally.

The result of the validation process can be viewed from the status of the inbound e-document. If the inbound e-document passed validation, then parsing and conversion will proceed. However, if the

inbound e-document is invalid, conversion will not proceed and the status of the inbound e-document will be set to Conversion Failed. The audit trail of the inbound e-document will indicate the details of a failed local conversion process or information from a validating third party.

Creating a Custom Plug-in for Inbound E-Document Validation

A custom plug-in implementation for validating inbound e-documents must be created so that it will be available in the vendor e-document package.

A validation custom plug-in implementation script is included in Electronic Invoicing SuiteApp. Use the sample script as a reference or a template for creating your own custom validation plug-in.

The validation plug-in must be a JavaScript file that is compatible with SuiteScript 2.0.

To open the sample validation plug-in script, go to Documents > Files > File cabinet > SuiteBundles > Bundle 436209 > src > comp > pl > pl_inbound_validation_sample.js. Open the script in a programming editor to view the script.

Note: This sample script uses the `define` function, which is required for an entry point script (a script you attach to a script record and deploy). You must use the `require` function if you want to copy the script into the SuiteScript Debugger and test it. For more information, see the help topic SuiteScript Debugger SuiteScript Debugger.

The following code is a sample validation plug-in script.

```
/**
 * Copyright (c) 2017, Oracle and/or its affiliates.
 *
 * @NModuleScope public
 * @NApiVersion 2.x
 * @NScriptType plugintypeimpl
 */

define([], function() {
    /**
    * validate - This function is the entry point of our plugin script
    * @param {Object} plugInContext
    * @param {Object} plugInContext.eDocument
    * @param {String} plugInContext.eDocument.id
    * @param {String} plugInContext.eDocument.scriptId
    * @param {String} plugInContext.eDocument.content
    * @param {Object} plugInContext.eDocument.source
    * @param {String} plugInContext.eDocument.source.id
    * @param {String} plugInContext.eDocument.source.text
    * @param {Object} plugInContext.eDocument.template
    * @param {String} plugInContext.eDocument.template.id
    * @param {String} plugInContext.eDocument.template.text
    * @param {Object} plugInContext.eDocument.status
    * @param {Integer} plugInContext.eDocument.status.id
    * @param {String} plugInContext.eDocument.status.text
    * @param {Object} plugInContext.eDocument.package
    * @param {String} plugInContext.eDocument.package.id
    * @param {String} plugInContext.eDocument.package.text
    * @param {Object} plugInContext.eDocument.transactionType
    * @param {String} plugInContext.eDocument.transactionType.id
    * @param {String} plugInContext.eDocument.transactionType.text
    * @param {Object} plugInContext.eDocument.vendor
    * @param {String} plugInContext.eDocument.vendor.id
    * @param {String} plugInContext.eDocument.vendor.text
    * @returns {Object} result
    * @returns {Boolean} result.success
    * @returns {String} result.message
    */
     function validate(pluginContext) {
```

```
        var eDoc = pluginContext.eDocument;
        var result = {
                success: false,
                message: ""
        };


        // Connect to validation service


        // If successful
        result.success = true;
        result.message = "Validation successful!";


        // Sample result if not successful
        // result.success = false;
        // result.message = "Service returned a failed response";


        return result;
    }


    return {
        validate: validate
    };

});
```

Important: The validation custom plug-in script must have the `@NSScriptType plugintypeimpl`.

You can create as many validation plug-ins implementation as needed in your e-document processing, but you can only assign one validation plugin implementation per e-document package.

After creating the custom validation plug-in script, upload it to Customization > Plug-ins > Plug-in Implementations > New . The type of the custom plug-in implementation must be “Validation Plugin”. For more information, Custom Plug-in Creation.

Setting Up the Inbound E-Document Validation Plug-in

Before inbound e-document processing can use the validation plug-in you created, you must set up the plug-in first and then integrate it with an e-document package.

To set up the inbound e-document validation custom plug-in and integrate it with a vendor e-document package:

1. Go to Setup > E-Documents > Inbound E-Document Validation Plugin > New.
2. On the Inbound E-Document Validation Plugin page, enter the name of the plug-in in the **Name** field. In the **Script** field, select the plug-in you created from the dropdown list.
3. Click **Save**.
4. Go to Setup > E-Documents > E-Document Package.
5. On the E-Document Package List page, click the Edit link of the vendor e-document package that you want to integrate the inbound e-document validation plug-in with.
6. On the E-Document Package page, in the **Inbound E-Document Validation Plugin Implementation** field, select the custom plug-in you created.
7. Click **Save**.

The validation process or logic that you programmed into the plug-in will now be integrated with the inbound e-document conversion process.

Deploying Automatic Bulk Conversion Script for Inbound E-Documents

The administrator can create schedules to run periodic and automatic bulk conversion of all inbound e-documents of which status is For Conversion. The Electronic Invoicing SuiteApp includes a Script Deployment for running automatic bulk conversion of inbound e-documents. By default, the status of the script is set to **Not Scheduled**. The administrator can set a schedule for the script to convert inbound e-documents in bulk.

Deploy the bulk conversion script for inbound e-documents:

1. Go to Customization > Scripting > Script Deployments.
2. Expand the Filters and select **Map/Reduce** in the Type field. Click the Edit link of the script **Convert Inbound E-Document MR**.
3. Clear the **Deployed** box if you do not want to deploy the script yet. A script will not run in NetSuite until the Deployed box is selected.
4. Select a status in the **Status** field:
   - Testing
   - Not Scheduled
   - Scheduled

   For more information, see the help topic Setting Script Deployment Status.
5. Choose an event type on the Schedule subtab:
   - **Single Event** – The script converts inbound e-documents only one time.
   - **Daily Event** – Enter the interval between days if this event should occur every day or every few days, or select every weekday if this event should occur every day except Saturdays and Sundays. For example, enter 1 as the interval if this event should occur every day, or enter 2 if the event should occur every other day.
   - **Weekly Event** – Enter the interval between weeks, and select the day of the week this event should be repeated.
   - **Monthly Event** – If you want to convert inbound e-documents on the same day of every month or every few months, enter the date when you want inbound e-document conversion to repeat and then select the interval between months. If you want to convert inbound e-documents on the same day of the week every month or every few months, select the week, the day of the week, and enter the interval between months.
   - **Yearly Event** – If you want to convert inbound e-documents one time every year, select the month and day, or select the week, day, and month.
   - **Start Date** – You must enter the date when you want to schedule inbound e-document conversion.
   - **Start Time** – Enter the time when you want inbound e-document conversion to start.
   - **Repeat** – Select how often you want the script to convert inbound e-documents. On the day the script is scheduled to run, inbound e-documents will be converted at the specified Start Time and the process repeats every n hours until midnight.
   - **End By** – Set the date when the script should stop converting inbound e-documents.

If you are scheduling a single event, inbound e-documents will be converted on the date entered in the Start Date field.

If you are scheduling a repeat event, inbound e-documents will be converted according to the schedule you set, starting from the date entered in the Start Date field and ending on the date entered in the End By field.

- **No End Date** – Check this box if you want to repeat the schedule indefinitely.

6. Click **Save and Execute** to convert inbound e-documents immediately. Click **Save** to convert inbound e-documents according to the schedule that you set.

> **Note:** Before deploying the bulk conversion script, make sure that the date format in inbound e-documents is the same as the date format set in Setup > Company > Preferences > General Preferences.

After running the automatic bulk conversion, all inbound e-documents with status For Conversion, will be converted into transaction records. Successfully converted inbound e-documents will have their status set to Converted; otherwise, the status will be Conversion Failed.

If any of the inbound e-documents in the batch failed conversion, a notification will be sent to the email address of the user defined in the Recipient of E-Document Notifications. If all inbound e-documents in the batch were successfully converted, no notification email will be sent. The email notification will have an attached CSV file listing the inbound e-documents that underwent conversion. The list has columns indicating the Internal ID, Vendor, and details of the conversion process and errors. Likewise, you can view the error details of an inbound e-document that failed conversion by clicking the View link of that record. Then, on the E-Document Audit Trail subtab, in the Details column, view the error scope and details.

Electronic Invoicing User Guide

> **Note:** To fully use the outbound and inbound e-document processing of the Electronic Invoicing SuiteApp, make sure your account administrator has already created and applied the e-document templates, sending method, email recipients, email capture, inbound validation plug-in, and other settings. For information about setting up Electronic Invoicing, see Electronic Invoicing Administrator Guide.

The following topics are intended for users of the Electronic Invoicing SuiteApp.

General User Task

- Displaying the E-Documents Portlet on the Home Page

Outbound E-Document Processing User Tasks

- Overview of Outbound E-Document Process
- Outbound E-Document Statuses
- Assigning E-Document Packages to Customer or Vendor Records
- Defining E-Document Email Recipients
- Selecting E-Document Packages, Templates and Sending Methods on Transactions
- Enabling PDF File Reference Generation
- Generating and Regenerating E-Documents
  - Generating E-Documents for Single Transactions
  - Regenerating E-Documents for Single Transactions
  - Generating and Regenerating E-Documents in Bulk

- Sending and Resending E-Documents
  - Sending the E-Document of a Single Transaction
  - Resending the E-Document of a Single Transaction
  - Resending E-Documents in Bulk

Inbound E-Document Processing User Tasks

- Overview of Inbound E-Document Processing
- Inbound E-Document Statuses
- Receiving Inbound E-Documents by Email Capture
- Receiving E-Document XML Files from Web Service
- Uploading Received XML Files as Inbound E-Documents
- Converting Inbound E-Documents into Transaction Records
  - Converting an E-Document into Vendor Bill Linked to Purchase Order
  - Prerequisites and Conditions for Conversion
- Common Scenarios in Vendor Bill Conversion
- Converting Individual Inbound E-Documents into Vendor Bills
- Converting Failed Inbound E-Documents
- Compatibility of Approval Workflows with Vendor Bill Conversion
- Canceling Inbound E-Documents

See also the following topics:

- Electronic Invoicing Overview
  - Understanding E-Documents and E-Document Packages
  - Electronic Invoicing Permissions and Access Levels
  - Electronic Invoicing Limitations and Best Practices
- Electronic Invoicing Errors
  - Electronic Invoicing Error Codes
  - Outbound E-Document Generation Errors
  - Outbound E-Document Sending Errors

Displaying the E-Documents Portlet on the Home Page

The Electronic Documents Dashboard SuiteApp Portlet provides links to outbound and inbound e-document processes and features. For more information, see Electronic Documents Dashboard SuiteApp Portlet

If you have access permission to the e-documents portlet, you can display it on your NetSuite Home page like any dashboard portlet.

To display the E-Documents Portlet on the Home page:

1. Go to the NetSuite Home page.
2. Click **Personalize**.
3. On the Personalize Dashboard menu, select **SuiteApps**.
4. Click the Electronic Documents portlet, or drag the icon to the dashboard.

For more information about dashboard portlets, see the help topic Dashboards.

Overview of Outbound E-Document Process

From a user's perspective, the following steps describe the end-to-end process of generating XML or JSON e-documents from NetSuite transaction records and then sending the e-documents:

1. Assign an e-document package to a customer or vendor by performing the following steps:
    a. Create or edit a customer or vendor record. If you are using an e-document sending method with an email channel, be sure to create or add the email recipients of the e-document. Then, on the **E-Document Email Recipient** subtab under the **E-Document** subtab, select contacts. You can select only contacts associated with the customer or vendor. Contacts must have valid email addresses. See Defining E-Document Email Recipients.
        - If the customer or vendor is a company, create one or more contact records.
        - If the customer or vendor is an individual, add a valid email address on the record.
    b. On the **E-Document** subtab, in the **E-Document Packages** field, select the e-document package to use for the customer's or vendor's transaction records.

        Note: The outbound template and sending method included in the e-document package you will select must have subsidiaries belonging to this customer or vendor. For more information, see Multi-subsidiary Support in the Outbound Process.
2. Create or edit a transaction record. Select an e-document template and an e-document sending method. For more information, see Transactions and Processes Supported by the Electronic Invoicing SuiteApp and Selecting E-Document Packages, Templates and Sending Methods on Transactions.
    - If the e-document to be generated must be sent to a certification authority for certification, make sure that a certification sending method is assigned to the transaction and subsidiary. For more information, see E-Document Certification in the Outbound Process.
3. Generate an e-document for the transaction record. You can generate e-documents individually or in bulk. For more information, see Generating and Regenerating E-Documents.
4. Send the e-document. You can send e-documents individually or in bulk. For more information, see Sending and Resending E-Documents.

    Note: If a certification sending method is assigned to the transaction and subsidiary, the Certify E-Document button is displayed, which sends the e-document to the specified certification authority. Certified e-documents can be sent again to customers, vendors or a tax agency.
5. If an administrator enables and configures the automatic e-document processing feature, you can process the e-document automatically by clicking the **Process E-Document** button. For more information, see Processing Multiple E-Document Processes Using Automatic E-Invoicing

To understand errors that can be encountered during the generation or sending of e-documents, read the following topics:

- Outbound E-Document Generation Errors
- Outbound E-Document Sending Errors

To understand the outbound electronic invoicing process flow and e-document status definitions, see the following topics:

- Outbound Electronic Invoicing
- E-Document Audit Trail and Statuses

Outbound E-Document Statuses

E-document statuses are indicated in the **E-Document Status** field on the **E-Document** subtab. Outbound e-document statuses describe whether a transaction or e-document is in the process of e-document generation, sending or certification. The following table lists the e-document generation statuses.

E-Document Generation Status	Description
For Generation	The e-document can be generated. An outbound e-document template was associated with the transaction and the e-document can be generated. Audit trail log indicates Tagged for Generation.
Generating	The e-document is being generated.
Generation Failed	The e-document was not generated. An error caused generation to fail. See details of e-document generation errors on the E-Document Audit Trail subtab. You must first fix the errors before you can regenerate an e-document. For more information, see Outbound E-Document Generation Errors.
Untagged for E-Invoice Generation	The e-document template was removed from the transaction record. The e-document of the transaction record will not be generated.

The following table lists the e-document sending statuses.

E-Document Sending Status	Description
Ready for Sending	The e-document was generated successfully and the e-document can be sent to specified recipients. Whether the e-document was generated manually or through bulk generation is indicated.
Sending	The e-document is being sent to specified recipients.
Sending Failed	The e-document was not sent. Information about sending errors are shown in the Details column. You must fix the error and regenerate the e-document before you can resend it.
Sent	The e-document was successfully sent. If the default sending method is email, the Details column shows the email addresses of the sender and recipients. **Note:** When an e-document is sent, the designated e-document sender's name and email address is displayed as the sender. If you did not designate a sender, the system uses the name and email address of the person who initiated the sending process as the sender of the e-document. For more information, see Selecting a Designated E-Document Sender.

The following table lists the e-document certification statuses.

Note: E-document certification statuses are not automatically updated by the system, you must implement the updates through the certification sending method. See Updating E-Document Certification Statuses

E-Document Certification Status	Description
Ready for Certification	The e-document has a certification sending method assigned to its transaction and subsidiary, and it can be sent to the certification authority.

E-Document Certification Status	Description
Certification in Progress	The e-document has been sent to the certification authority and is being certified. The certified e-document has not been received.
Certification Failed	Certification failed due to an error, of which details are on the E-Document Audit Trail subtab. You must fix the error and regenerate the e-document before you can send it again for certification.
Certification Data Error	Details of certification errors are on the E-Document Audit Trail subtab.
Ready for Sending	The e-document was successfully certified and is ready for sending.

Assigning E-Document Packages to Customer or Vendor Records

The e-document templates and sending methods assigned to the e-document package are available for selection on the **E-Document** subtab of every customer or vendor transaction.

To assign an e-document package to a customer or vendor:

1. Create or edit a customer or vendor record.
2. Click the **E-Document** subtab of the customer or vendor record.
3. In the **E-Document Package** field, select the appropriate e-document package for the customer or vendor.

   Note: The outbound template and sending method included in the e-document package you will select, must have subsidiaries belonging to this customer or vendor. For more information, see Multi-subsidiary Support in the Outbound Process.
4. Click **Save**.

Note: Assigning e-document packages with email sending methods can only be done by editing the customer or vendor record.

Defining E-Document Email Recipients

If you want to use an e-document sending method that has an email sending channel, you must first define the email recipients for your customer or vendor. If the customer or vendor is a company, there should be at least one email recipient defined on the **E-Document** subtab on the customer or vendor record. If the customer or vendor is an individual, make sure it has an email address.

Note: The system can send an e-document by email to a maximum of 10 recipients for each customer or vendor. The system counts each contact added as a recipient. If you add the same contact multiple times, each instance is considered an individual recipient.

To define e-document email recipients:

1. Create or edit a customer or vendor record.
2. On the **E-Document Email Recipient** subtab under the **E-Document** subtab, select a contact or select **New** to create a new contact.

> **Note:** When you select an email recipient, only contacts with valid email addresses associated with the customer or vendor are available for selection in the dropdown list.

3. Click **Add** after selecting each contact.
4. Click **Save**.

> **Note:** For transfer order, basic and journal style custom transaction types, you can define the email recipients directly in the **E-Document Email Recipient** field from the **E-Document** subtab. The email recipients are also applicable for item fulfillment transactions.

Selecting E-Document Packages, Templates and Sending Methods on Transactions

Before you can generate an e-document, you must specify on the transaction record the e-document template and sending method to use for generating and sending the e-document.

To Select E-Document Package, Template and Sending Method on Transactions:

1. Create or edit any supported transaction record that you want to generate and send an e-document.
2. Go to the **E-Document** subtab.
3. Select an e-document package associated with the e-document template and sending method on the **E-Document Package** field.

   For item fulfillment records, you must first define an Item Fulfillment template and sending method along with Transfer Order template and sending method in the same package. This is to enable the selection of Item Fulfillment template and sending method directly from the Item Fulfillment record.

   > **Note:** The **E-Document Package** field is specific only for transfer order, its following item fulfillment, basic and journal style custom transactions types.

4. Select an e-document template that you want the system to use for generating the e-document on the **E-document Template** field.

   If you do not want the transaction to be tagged for e-document generation, leave this field blank.

   > **Note:** The e-document templates available for selection are based on the e-document package assigned to the customer. For more information, see Creating E-Document Packages

5. Select the sending method to use for sending the e-document on the **E-Document Sending Method** field.
6. Click **Save**.

Enabling PDF File Reference Generation

The Electronic Invoicing SuiteApp supports the creation, sending, and receiving of PDF file references of transaction records undergoing the e-document process. The PDF version of transaction records will enable you to view and check if the content of e-document is correct.

In outbound e-document processing, if you enable the setting Generate PDF, the PDF file is created when the XML or JSON file of a transaction record is generated. After the PDF file is created, you can view it in the **Generated PDF** field, on **E-Document** subtab of transaction records. When sending the e-document using **E-document Sending Method Plugin Implementation** field, the PDF file is attached along with the XML or JSON e-document.

Note: PDF file reference generation is not supported for vendor bill records. The Generate PDF box and Generated PDF link are not available on vendor bill records with generated outbound e-documents.

Generate PDF File References

You can generate PDF file references of transaction records to be converted into e-documents.

The PDF file generated will follow the default printing layout of the transaction. You can set a default preference on entity records.

Enabling Generate PDF as a Default in Entity Records

1. Edit a customer or vendor record and go to the **E-Document** subtab.
2. Check the **Generate PDF** box. Consequently, all transaction types supported by electronic invoicing from this customer or vendor will have the Generate PDF box automatically checked. When e-documents are generated from the transactions, corresponding PDF file references will be automatically created.When sending the e-document using **E-document Sending Method Plugin Implementation** field, the PDF file is attached along with the XML or JSON e-document.
3. Click **Save**.

   When the e-document is generated, the PDF file reference can be viewed and downloaded under **Generated PDF** on the **E-Document** subtab of the transaction. Also do note that the QR code will be visible only on successful QR generation, else it will be hidden.

If you do not want to create PDF file references of individual transaction records, you can still override PDF generation enabled on the customer or vendor record. To do so, edit the transaction record that you do not want to generate a PDF file reference of, and then clear the **Generate PDF** box. If Generate PDF is cleared on a transaction record where it was initially enabled and a PDF file was generated, during the next e-document generation, no PDF file will be generated and any previously generated PDF file will be deleted.

Generating and Regenerating E-Documents

The e-documents of transaction records in NetSuite can be generated using the Electronic Invoicing SuiteApp. The generated e-documents can then be sent to your customers or tax authorities through the outbound process.

You can generate the XML or JSON e-documents of the following transaction records:

- Bill (for self-billing)
- Cash Sale
- Cash Refund
- Credit Memo
- Customer Payment
- Estimate

- Invoice
- Item Fulfillment
- Purchase Order
- Return Authorization
- Transfer Order
- Vendor Credit or Bill Credit

To generate an e-document, you must first select an e-document template on the transaction. The system cannot generate an e-document if there is no e-document template associated with the transaction.

The e-document templates available for selection in each transaction record are templates you have assigned to the customer's e-document package. For more information, see Creating E-Document Packages.

The e-document templates available for selection in each transaction record are templates that have been assigned to that transaction type. For more information, see Creating E-Document Templates.

You can generate e-documents individually or in bulk, regardless of transaction type.

The system supports e-document generation for supported transactions created using the user interface, CSV import, SOAP web services, and SuiteScript.

Generating E-Documents from the Electronic Documents Portlet

You can quickly access outbound e-documents with status For Generation from the Electronic Documents portlet on the Home page. For more information about the Electronic Documents portlet, see Electronic Documents Dashboard SuiteApp Portlet.

To generate e-documents from the Electronic Documents portlet:

1. Go to the Electronic Documents portlet on the Home page.
2. Click the number under **Outbound E-Documents for Generation**.

   The Outbound E-Documents For Generation results page is displayed.
3. Click the **View** link of the transaction record that you want to generate an e-document from.
4. On the selected transaction record, click the **Generate** button.

Generating E-Documents for Single Transactions

You can also start the process of generating e-documents from transaction records.

To generate an e-document:

1. Create or edit a transaction record.
2. Enter items and other required information about the transaction record.
3. Click the **E-Document** subtab.
4. In the **E-Document Template** field, select the e-document template that you want the system to use for generating the e-document. If you do not want this transaction to be tagged for e-document generation, leave this field blank.

Note: The e-document templates available for selection are based on the e-document package assigned to your customer or vendor. For more information, see Creating E-Document Packages.

Important: Use caution when you want to edit the transaction record. Each time you edit and save a transaction, the **E-Document Status** field is updated to **For Generation**. You will need to generate and send the e-document again. Note that this does not apply to e-documents with the **Sent** status.

5. Click **Save**. The system updates the transaction record, and a **Generate E-Document** button is displayed at the top. On the **E-Document** subtab, the value of the **E-Document Status** field is updated to **For Generation**.

   Note: The **Generate E-Document** button appears only when viewing a transaction record. It is not displayed in edit mode.

6. Click **Generate E-Document**.

   The outbound e-document is generated. A PDF version of the source transaction is also generated if Generate PDF is enabled on the transaction record.

   After generating the e-document, the system updates the value in the **E-Document Status** field on the **E-Document** subtab. The field shows one of the following statuses:

   - **Ready for Sending** – This status means the e-document was generated successfully and can be sent to the email recipients defined in the customer record or your e-document sending method.

     For more information, see Creating E-Document Sending Methods.
   - **Generation Failed** – This status means the e-document was not generated. Information about generation errors are shown in the **Details** column of the **E-Document Audit Trail** subtab. You must first fix the errors before you can regenerate an e-document.

   For more information, see E-Document Audit Trail and Statuses.

Regenerating E-Documents for Single Transactions

You can regenerate an e-document for individual supported transaction records at any time. The transaction records must have an e-document template assigned to them, and the status in the **E-Document Status** field must not be blank or **Sent**. However, if an error occurs during generation and the error is not fixed, the system will not be able to generate an e-document for the transaction.

If the value in the **E-Document Status** field on the **E-Document** subtab shows **Generation Failed**, you can regenerate the e-document by doing the following:

To regenerate an e-document:

1. Fix the errors identified in the audit trail.
2. View the transaction and click **Generate E-Document**.

Important: Use caution when you want to edit the transaction record. Each time you edit and save an document, the **E-Document Status** field is updated to **For Generation**. You will need to generate and send the e-document again. This can result in generating and sending multiple e-documents for one transaction record. Note that this does not apply to e-documents with the **Sent** status.

For more information about generation errors, see Outbound E-Document Generation Errors.

Generating and Regenerating E-Documents in Bulk

If your account administrator has set up a schedule to run the bulk generation script periodically, the script automatically generates e-documents for those transactions that have any of the following statuses:

- For Generation
- Generation Failed

Upon successful bulk generation of outbound e-documents, PDF versions of the transactions are also generated if those transactions have the Generate PDF setting enabled.

For failed generation, errors must be fixed first before e-documents can be regenerated.

When the system encounters an error during bulk generation of e-documents, it sends an email notification to the Recipient of E-Document Notifications. If the Recipient of E-Document Notifications is not assigned yet, the notification will be sent to the active administrator of the account. For more information about assigning the Recipient of E-Document Notifications, see Prerequisites for Using Electronic Invoicing, step 4.

The email message contains an attachment in CSV format that lists the transaction records that encountered errors during bulk generation.

The user who created the transaction record must fix the errors for each transaction record.

After the errors are fixed, the e-documents will be generated during the next scheduled bulk generation of e-documents. Alternatively, the user can regenerate an e-document for the individual transaction record.

For more information about generation errors, see Outbound E-Document Generation Errors.

If you want to generate or regenerate e-documents for multiple transactions outside of the scheduled script run, contact your account administrator for assistance. See Deploying the Bulk Generation Script for E-Documents.

Sending and Resending E-Documents

A successfully generated e-document displays the **Send E-Document** button on the transaction record and its e-document status indicating Ready for Sending.

A generated e-document with a certification sending method assigned to its transaction and subsidiary, displays the **Certify E-Document** button on the transaction.

Note: Generating a new e-document overwrites the previous e-document.

Sending the E-Document of a Single Transaction

Perform the following steps to send the e-document of a single transaction.

To send an e-document:

1. Open the transaction record whose e-document status is Ready for Sending.

   If the e-document is for certification and a certification sending method is assigned to the transaction and subsidiary, the status is Ready for Certification.
2. Make sure that the **E-Document Sending Method** field specifies the sending method appropriate for this transaction record.

If the e-document is for certification, make sure that a certification sending method is assigned to the transaction and subsidiary.

3. Click the **Send E-Document** button.

   If the e-document is for certification and a certification sending method is assigned to the transaction and subsidiary, click the **Certify E-Document** button.

A banner appears on the transaction record indicating that sending of the e-document is in progress. The e-document is sent through the selected sending method and the value of the E-Document Status field on the E-Document subtab is updated. For more information, see Outbound E-Document Statuses.

Successfully certified e-documents are returned by the certification authority. You can view or download the XML or JSON file of the certified e-document by clicking the corresponding links under **Certified E-Document** on the **E-Document** subtab of the transaction.

Resending the E-Document of a Single Transaction

Perform the following steps to resend the e-document of a single transaction.

To resend an e-document:

1. If failure in the previous e-document sending was caused by an error, amend the transaction as necessary and save it.
2. Open the amended transaction and click **Generate E-Document**.

   After the system generates the e-document, an audit trail is created and the status in the **E-Document Status** field changes to **Ready for Sending**.
3. Click **Send E-Document** to send the e-document.

   After the system sends the e-document, the status displayed in the **E-Document Status** field changes to **Sent**, and an audit trail is created.

Resending E-Documents in Bulk

Note: Resending e-documents in bulk requires certain permissions and access levels for roles. For more information, see Setting Up Custom Roles to Send E-Documents.

The Electronic Invoicing SuiteApp provides a search filter to enable you to resend in bulk the e-documents that failed initial sending. The system searches for all e-documents with Sending Failed status.

Errors must be fixed before e-documents can be successfully resent. For more information, see Outbound E-Document Sending Errors.

To access e-documents that failed generation or sending, go to the Electronic Documents portlet on the Home page, and then click the number under **E-Documents with Errors**. On the E-Documents with Errors results page, view or edit the transaction records that you want to fix.

After fixing the errors, you can proceed with either regenerating or resending the e-documents.

To resend e-documents in bulk:

1. Go to the Electronic Documents portlet on the Home page, and then click the link **Send Failed E-Documents**.
2. In the Select entity type filter, choose either **Customer** or **Vendor**.

3. To view a list of transactions for a particular customer or vendor, select that customer or vendor from the **Customer** or **Vendor** field. If no customer or vendor is selected, the search result will display all transaction belonging to the subsidiary, regardless of customer or vendor.
4. In the **Subsidiary** field, select a subsidiary to view the transactions for that subsidiary only.
5. In the **Transaction Date From** field, set the date to define the beginning of the date range. The search result displays all created transaction records starting from the date you set in this field.
6. In the **Transaction Date To** field, set the date to define the end of the date range. The search result displays all created transaction records starting from the date you set in this field.
7. If you chose the Customer entity type, select one or more transaction types in the **Transaction Type** field, for each e-document you want to resend. To select multiple transaction types, press and hold the Ctrl key while selecting each transaction type. If no transaction type is selected, the search result will show all e-documents with Sending Failed status, regardless of transaction type.

   If you chose the Vendor entity type, Purchase Order is automatically selected in the Transaction Type field, because only purchase order is applicable to vendor entities.

   Only transaction types that are supported by or applicable to outbound e-document sending are shown in the transaction list, which include:

   - Bill (for self-billing)
   - Cash Sale
   - Cash Refund
   - Credit Memo
   - Customer Payment
   - Estimate
   - Invoice
   - Item Fulfillment
   - Purchase Order
   - Return Authorization
   - Registered Custom Transaction Types
   - Transfer Order
   - Vendor Credit or Bill Credit
8. Click **Search**.

   The search result displays all e-documents with Sending Failed status, according to your search criteria.
9. Click **Send** to resend all e-documents displayed in the search results.

   After the system sends the e-documents, it updates the value in the **E-Document Status** field on the **E-Document** subtab on each transaction record. The field shows one of the following statuses:

   - **Sent** – This means the e-document was successfully sent. The **Details** column shows the email addresses of the sender and recipients.
   - **Sending Failed** – This means the e-document was not sent. Information about sending errors are shown in the **Details** column. You must fix the errors before you can successfully resend the e-document.

   For more information, see Outbound E-Document Sending Errors.

   The system sends an email notification to the person who initiated the sending process when the e-documents have been sent.

   The email message contains an attachment in CSV format that lists the transaction records and the status of each transaction. If errors were encountered, the details are included in the attachment.

Processing Multiple E-Document Processes Using Automatic E-Invoicing

Only users with administrator role can enable and configure the e-document processes for automation. After the configuration, only users with required permissions can get access to the **Process E-Document** button on individual transactions. After this the users can perform the pre-configured e-document processes automatically for a subsidiary by clicking the **Process E-Document** button .

E-Document Network Status

The Electronic Invoicing SuiteApp enables you to get the e-documents certified from a regulatory body and get network status through an API. The latest network status can be obtained by clicking the **Get Network Status** button on the transaction.

The **Get Network Status** button is available on the transaction as soon as the sending method returns the network details for the transaction. It can happen either in the certify or send stage of the transaction, or both. If the transaction is in certify stage, the certification sending method returns the status of the transaction, else if the transaction is in send stage, the sending method assigned to the transaction returns the status of the transaction.

The following table describes the transaction body fields which contain the network information.

Network Field	Field ID	Description
Network Reference Id	custbody_ei_network_id	It is a unique number generated by network API while certifying or sending e-document.
Network Name	custbody_ei_network_name	It is the name of a network. For instance, PEPPOL, SAT, ARIBA.
Network Status	custbody_ei_network_status	It displays a keyword representing one of the possible network statuses used by the network.
Network Status Updated On	custbody_ei_network_updated_date_time	It displays the time stamp (date and time) of the updated status.

Overview of Inbound E-Document Processing

The following describe the end-to-end process of creating inbound e-documents and then converting them into NetSuite transaction records using the Electronic Invoicing SuiteApp.

To upload and convert inbound e-documents:

1. The administrator creates an inbound e-document template, which will map the data elements of the inbound XML e-document to corresponding NetSuite transaction fields when converted. See Creating E-Document Templates.
2. The administrator associates the inbound e-document template with an e-document package, which is assigned to a customer or vendor. See Assigning E-Document Packages to Customer or Vendor Records.
3. The administrator designates an employee or group who will receive an email notification after scheduled automatic bulk conversion of inbound e-document records to transaction records.
4. If the inbound e-document is received through means other than email capture, users can upload it manually. See Uploading Received XML Files as Inbound E-Documents.

5. Users can convert inbound e-document records with status For Conversion, into transaction records. You can manually convert e-documents or run automatic scheduled bulk conversion. See Converting Inbound E-Documents into Transaction Records.

   The status of newly created vendor bills from e-document conversion is dependent on the default status of Accounting Preference or whether Vendor Approval Routing is enabled.

Inbound E-Document Statuses

E-document statuses are indicated in the **E-Document Status** field on the **E-Document** subtab. Inbound e-document statuses describe the state of a received e-document in the e-document conversion process. The following table lists the inbound e-document statuses.

E-Document Conversion Status	Description
Incomplete	The inbound e-document is missing some details or content. Audit trail log indicates Tagged as Incomplete.
For Conversion	The inbound e-document is ready for conversion into a vendor bill. A vendor, XML file reference, and e-document template have been defined in the inbound e-document. Audit trail log indicates Tagged for Conversion.
Converting	The e-document is being converted to a transaction record.
Converted	The inbound e-document was successfully converted into a vendor bill.
Conversion Failed	An error occurred and the inbound e-document was not converted. Information about conversion errors can be found in the Details column of the E-Document Audit Trail subtab.
Canceled	The inbound e-document was canceled. If cancellation of the inbound e-document encountered an error and did not proceed, the audit trail log will indicate Cancellation Failed.

Receiving Inbound E-Documents by Email Capture

After your administrator has set up the Inbound Email Capture plug-in and Vendor E-Document Sender, you are ready to receive inbound e-documents from your vendors. The system automatically receives inbound e-documents sent through email and then queues them as Inbound E-Documents for Conversion if an e-document template is assigned.

The number of e-documents received from email capture is displayed on the Electronic Documents portlet, under Inbound E-Documents for Conversion. You can click the number displayed to go to the Inbound E-Documents for Conversion: Results page, which lists the inbound e-documents for conversion. For more information, see Electronic Documents Dashboard SuiteApp Portlet.

If you have automatic bulk conversion set up, the bulk conversion script will automatically convert the e-documents in the list into transaction records at a defined schedule. For more information, see Deploying Automatic Bulk Conversion Script for Inbound E-Documents. Or, you can convert individual e-documents in the list by clicking the View link of an e-document, and then click the Convert button on the inbound e-document record page.

Inbound e-document email capture receives XML files along with their PDF file reference attachment, if any were generated. Received PDF files are stored in the File Cabinet and can be viewed by clicking a link in the PDF File Reference field on the newly created inbound e-document record. When the inbound e-document is converted into vendor bill, the PDF file reference can also be viewed from the E-Document subtab.

Note: Inbound email capture can only receive one XML e-document and one PDF file per email.

Editing Incomplete Inbound E-Documents

Inbound e-documents that do not clearly define the vendor who sent them, will also be missing the e-document template, which is defined by the vendor. These inbound e-documents will be tagged as incomplete. They are queued as Incomplete Inbound E-Documents, which is displayed on the e-document dashboard portlet. You can click the number displayed under Incomplete Inbound E-Documents to go a results page, where you can click the Edit link of an inbound e-document to complete it.

On the Inbound E-Document record page, you can complete the inbound e-document by selecting the right vendor in the Vendor field and then the system automatically selects the e-document template that is applicable to the inbound e-document based on the XML content and the selected vendor. But, you can still manually select or override the default e-document template. After completing the inbound e-document, the E-Document Status is changed to For Conversion.

Receiving E-Document XML Files from Web Service

XML files sent through web service are received by NetSuite and are displayed in the Electronic Documents dashboard portlet. But before receiving e-documents, you must first set up and select the appropriate inbound e-document templates to be used. For more information about setting up inbound templates, see the help topics Understanding Inbound E-Document Templates in JSON Format and Understanding XSD in Inbound E-Document Templates.

XML e-documents received by NetSuite through web service are validated and checked if the information they contain is complete. The system automatically identifies the sender using the Token ID and Token Secret provided in the request. The Web Service ID is used to identify the vendor whom the e-document should be associated with. The integrity of the XML file is also checked. The results of these validations determine if an inbound e-document record will be created from the received XML file. If created, the inbound e-document record will indicate Web Service in the Source field. The inbound e-document is added to the queue corresponding its processing status displayed on the Electronic Documents dashboard portlet.

NetSuite responds to web service requests by confirming the success or failure of e-document creation. You can view web service requests and responses by going to Customization > Scripting > Scripts, then click the View link of the Inbound E-Document Web Service RL script. The logs are on the Execution Log subtab.

Web Service Errors

The following tables list the web service errors that can be encountered by the recipient and sender of web service requests. The recipient is the user in your company who can receive the XML files from web service. The sender is the vendor you authorized to issue web service requests to NetSuite to send XML files.

Errors Encountered by Web Service Recipients

Error Message	Description and Solution
The Web Service ID is already being used by another vendor. Enter a different Web Service ID.	This message is displayed when user tries to save a vendor record that has a web service identifier that is already being used by another vendor. A different web service identifier must be entered.

Error Message	Description and Solution
The inbound e-document is incomplete, as the correct template cannot be determined. Either select a template in the inbound e-document record, or set up the XSD in the e-document template record to enable template autoselection.	This message is sent through email to the Recipient of E-Document Notifications, informing the recipient that an inbound e-document record was created but is incomplete. Either manually set the template in the inbound e-document record, or set up the template's XSD to enable template auto selection.
The inbound e-document is incomplete, as the correct vendor cannot be determined. Either select a vendor in the inbound e-document record, or set the Web Service ID in the associated vendor record.	This message is sent through email to the Recipient of E-Document Notifications, informing the recipient that an inbound e-document record was created but is incomplete. Either manually set the vendor in the inbound e-document record, or correctly set up the vendor's web service ID field.

Errors Encountered by Web Service Sender

Error Message	Description and Solution
The following keys are missing: {KEYS}, which you must provide in the web service request.	The web service request failed because any of the following keys is missing: - identifier - filename - content Ensure that the missing keys are provided in the web service request.
The body of the web service request must be a JSON object or an of JSON objects using Content-Type: 'application/json'.	The web service request failed. The content of the web service request must be JSON objects.
The selected XML File Reference is not a valid XML file. Ensure that the file you select has the .xml extension.	The web service request failed. Provide a valid XML file for the XML File Reference.
No vendor is associated with the Web Service ID: {IDENTIFIER}. Ensure that the correct Web Service ID is used.	The web service request failed. Ensure that the Web Service ID is associated with a vendor or the party sending XML files.
The selected XML File Reference is not a well-formed XML document.	The web service request failed. Check the XML File Reference and ensure that content adheres to XML syntax, with properly defined content, structure and tags.

Uploading Received XML Files as Inbound E-Documents

Aside from email capture, you can receive vendor bills in XML format through other means like storage media. You can manually upload these XML file references to NetSuite as inbound e-document records, which are subsequently converted into vendor bills records.

To upload an XML file reference as an inbound e-document record:

1. On the E-Documents Portlet on the Home page, click the **Upload Inbound E-Document** link.

   The Inbound E-Documents page displays the following fields:

   - Transaction Type indicates the record (Bill) that will be created consequently of the conversion.
   - Source indicates Manual Upload by default.
   - Reference Number and PO Number receive values from the XML file reference that you will upload. These fields will have values after conversion to a transaction record.
   - E-Document Status indicates the status of the newly created inbound e-document.
2. In the **Vendor** dropdown list, select the vendor who sent the XML file.
3. In the **XML File Reference** field, click + and select the XML file that you will convert into a transaction record.

   The XML File Reference dropdown list displays files stored in the File Cabinet. Ensure that you select an XML document that is well-formed and valid, with the .xml file extension; otherwise an error will be generated.
4. In the **E-Document Template** field, select an inbound e-document template.

   Note: If an XSD file was selected in a template record that is appropriate for this inbound e-document record, that template is automatically selected in the E-Document Template field.
5. If you want to attach the PDF version of the XML document, click + on the **PDF File Reference** field, then select the PDF file you want to associate with this record.
6. Click **Save**.

A new inbound e-document record is created and displayed with its E-Document Status set to For Conversion. The **E-Document Audit Trail** subtab displays information about the inbound e-document record including the date created, entity, event type, owner, and details. If the uploaded XML file has an attached PDF file reference, the attachment can be viewed and downloaded in PDF File Reference field on the created inbound e-document record.

Converting Inbound E-Documents into Transaction Records

Converting an E-Document into Vendor Bill Linked to Purchase Order

The process of converting purchase order to vendor bill is viewed from the perspective of the NetSuite user being a customer, who purchased items or services from a vendor. In this scenario, the NetSuite user initially sends a purchase order to a vendor, by outbound electronic invoicing. The purchase order is received by the vendor and processed on their side. The vendor will enter the details of the purchase order into their system and eventually generates an invoice record, which is then converted into an XML e-document and returned to the NetSuite user. The XML e-document from the vendor is received by the NetSuite user through the inbound electronic invoicing feature. The NetSuite user uploads the received XML file to NetSuite as an inbound e-document record. It is this inbound e-document record that will undergo conversion to a vendor bill that is linked to the original purchase order.

Purchase Order Items and Expenses in Inbound E-Documents

As an item or an expense is required in a purchase order, either of them must also be included in the e-document for conversion; otherwise, conversion to vendor bill will not proceed.

In addition, items or expenses specified in the inbound e-document must include relevant details:

- For items, either the vendor code or vendor code/name must be included, depending on whether the Multiple Vendors feature is enabled or not.
- For expense, the amount must be included, and it is that the Default Expense Account in the vendor record has a value. If there are specified amounts with no corresponding accounts, the Default Expense Account is automatically made the default account in the new vendor bill created from conversion.

For more details about the prerequisites for converting inbound e-documents with purchase orders items or expenses, see the following topic.

Prerequisites and Conditions for Conversion

Permission to Convert E-Documents

Your user role must have the permission to convert e-document records into NetSuite transactions. For more information about granting a role the permission to perform inbound e-document conversion, see Setting Up Custom Roles that can Convert Inbound E-Documents.

Status of Purchase Order

The inbound XML e-document must not be linked to a purchase order whose status is fully billed, unapproved, rejected, cancelled or closed. Otherwise, the conversion will fail. Conversion will also fail if the vendor in the inbound XML e-document is different from the vendor in the purchase order.

Inbound E-Document has Reference Number with Mapping in the Template

The Reference Number must be included in the inbound XML e-document and the e-document template must contain the mapping for the reference number. Without the reference number in the inbound XML e-document or the correct reference number mapping in the template, conversion of the e-document will fail.

Inbound E-Document has Purchase Order Number and the Template has Mapping to the createdfrom Field

The Purchase Order Number must be included in the inbound XML e-document and the e-document template must contain the mapping for the createdfrom field. Without the Purchase Order Number in the inbound XML e-document or the correct reference number mapping in the template, conversion will result in a stand-alone vendor bill. See Converting an Inbound E-Document Without a Purchase Order Number.

Purchase Order must have either an Item or Expense

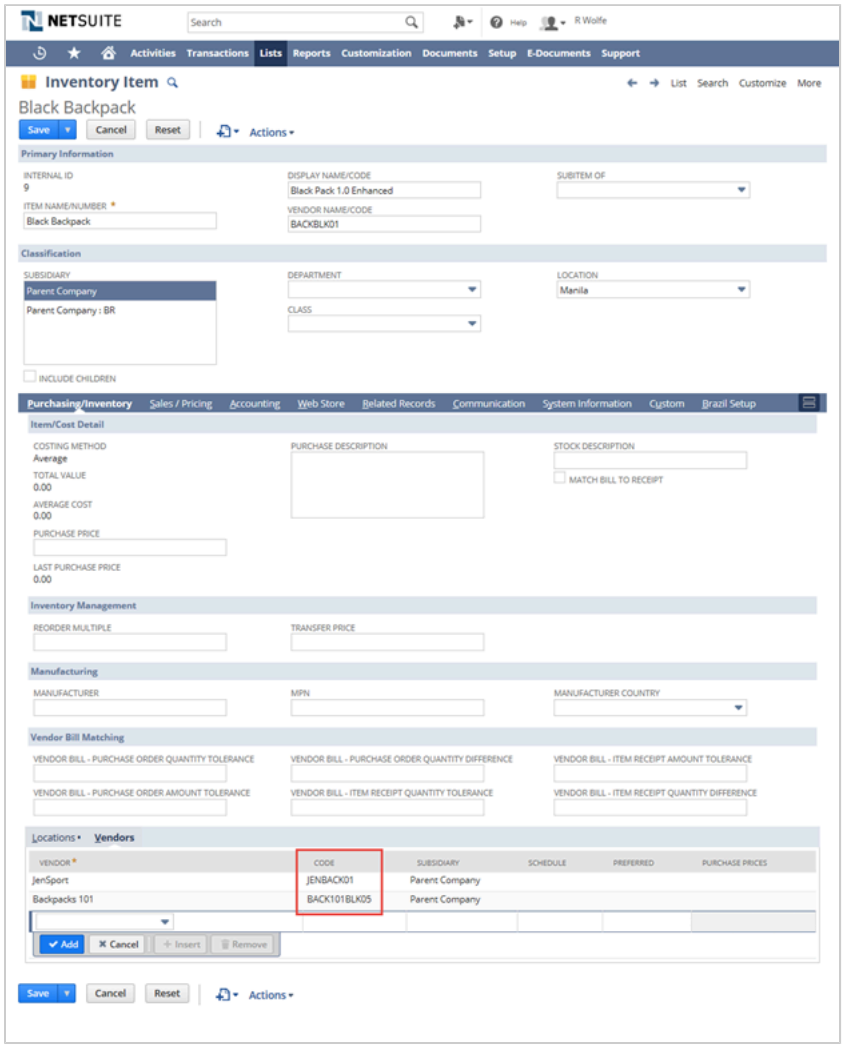
The reference purchase order of the inbound e-document for conversion, must have either an item or expense specified. Without item or expense, the inbound e-document will not be converted. Moreover, purchase order items or expenses must have required details included in the inbound e-document for conversion. The required details of items or expenses are discussed in the following sections.

Required Account and Amount for an Expense

If the XML e-document has an expense line, the required Account and Amount fields of the bill must have values. The Account field of the expense line references the value of Default Expense Account on the Financial subtab of the vendor record. Therefore, the vendor must have a Default Expense Account setup or the conversion will fail.

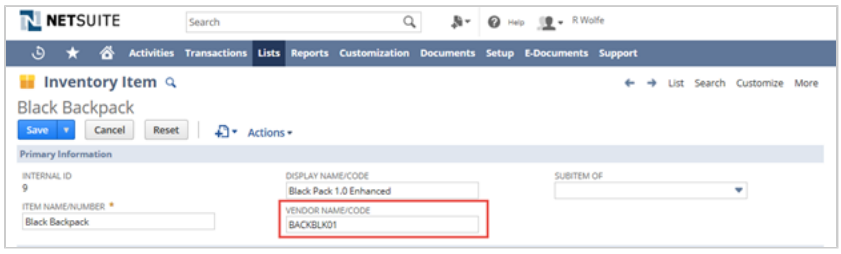
Required Vendor Code for Item Records

If the purchase order has item records, the vendor code of these item records must be included in the inbound XML e-document. If the Multiple Vendors feature is enabled during conversion, you must enter the Vendor Code on the Vendors subtab of the item record. The vendor code will be used to map the items to their corresponding vendor. If the vendor code in the XML e-document does not match any vendors, conversion will fail. The vendor code field is highlighted in the following screenshot.



Required Vendor Code/Name for Item Records

If the Multiple Vendors feature is disabled, the vendor code/name on the main tab of item records must be included in the inbound XML e-document. Otherwise, the conversion will fail. If two or more items have the same vendor code/name, implying duplicates, the conversion will fail. The vendor code/name field is highlighted in the following screenshot.



Common Scenarios in Vendor Bill Conversion

The following tables summarize the conversion of inbound e-documents with reference purchase orders that have items and expenses. Consider the scenarios that result in successful conversion and avoid those where conversion will fail.

Conversion Scenarios with Purchase Order Items

Case	Details of Reference Purchase Order	Details of Inbound XML E-Document based from the Purchase Order	Details of Inbound E-Document Template	Expected Conversion Result
Partial billing of some line items	The purchase order contains multiple line items.	The inbound XML e-document contains less line items than the total number of line items in the reference purchase order.	The template is valid with correct mapping.	Conversion will succeed. The created vendor bill includes only the line items contained in the XML e-document. The vendor bill is linked to the reference purchase order.
Duplicate line items	The purchase order contains duplicate line items.	The inbound XML e-document contains only one instance of each line item in the purchase order.	The template is valid with correct mapping.	Conversion will succeed. The created vendor bill contains no duplicate line items. The vendor bill is linked to the reference purchase order.
E-document template contains no mapping	The purchase order contains line items with values for amount, quantity, tax code, and rate fields.	The inbound XML e-document contains line items with field values.	The e-document template is valid but has no mapping for line item field values.	Conversion is successful. Field values for line items will take the default values entered in the purchase order.
The XML e-document contains additional line items that are not in the purchase order.	The purchase order is not yet billed and contains multiple line items.	The inbound XML e-document contains all or some of the purchase order line items, plus additional line items that are not in the purchase order.	The e-document template is valid with correct mapping.	Conversion will succeed. . All purchase order line items that are included the XML e-document, are entered in the created vendor bill. The additional line items included in the XML e-document but not in the purchase order, are also entered in the vendor bill. If all line items from the purchase order are included in the vendor bill. The status of the reference purchase order becomes Fully Billed.
Converting a Partially Received purchase order with additional line items that are not in the reference purchase order	The purchase order is Partially Received and contains multiple line items, one of which has been received.	The inbound XML e-document contains some (but not all) purchase order line items, including the ones that have been received, plus additional line items that are not in the purchase order.	The e-document template is valid with correct mapping.	Conversion will succeed. All purchase order line items in the XML e-document that are not yet received, are included in the created vendor bill. Line items already received but still included in the XML e-document, are taken as additional line items in the bill. Additional line items that are not in the purchase

Case	Details of Reference Purchase Order	Details of Inbound XML E-Document based from the Purchase Order	Details of Inbound E-Document Template	Expected Conversion Result
				order are included in the vendor bill. The status of the reference purchase order remains Partially Received.
The inbound XML e-document contains line items with no field values or values are set to 0.	The purchase order contains line items with values for amount, quantity, tax code, or rate.	The inbound XML e-document contains line items but with no values for the amount or quantity fields, or values are set to 0.	The e-document template is valid with correct mapping.	Conversion will fail. Details of the error are indicated in the E-Document Audit Trail.
The inbound XML e-document contains no line items.	The purchase order contains multiple line items.	The inbound XML e-document contains no line items.	The e-document template is valid with correct mapping.	Conversion will fail. Details of the error are indicated in the E-Document Audit Trail. The transaction must contain at least one line item.
The XML e-document contains only line items that are not in the reference purchase order.	The purchase order contains multiple line items.	All line items in the XML e-document are not in the reference purchase order.	The e-document template is valid with correct mapping.	Conversion will fail. Details of the error are indicated in the E-Document Audit Trail. The XML e-document must contain at least one line item from the reference purchase order.

Conversion Scenarios with Purchase Order Expenses

Case	Details of Reference Purchase Order	Details of Inbound XML E-Document based from the Purchase Order	Details of Inbound E-Document Template	Expected Conversion Result
The XML inbound e-document is for partial billing. Not all purchase order expenses are billed.	The purchase order contains expenses with the required amount and corresponding account.	The inbound XML e-document does not include all amounts for each expense line.	The e-document template is valid with correct mapping.	Conversion will succeed. The created vendor bill has the amount of the expenses mapped to their corresponding accounts.
An extra expense amount is included in the inbound e-document.	The purchase order contains expenses with the required amount and corresponding account.	The inbound XML e-document contains an expense amount with no corresponding account.	The e-document template is valid with correct mapping.	Conversion will succeed. The expense amount with no corresponding account will automatically be assigned to the Default Expense Account specified in the vendor record.
The purchase order has expense amount and account, but the inbound XML e-document does not	The purchase order contains expenses with the required amount and corresponding account.	The inbound XML e-document does not have an expense included.	The e-document template is valid with correct mapping.	Conversion will only proceed if the inbound XML e-document includes at least one line item from the reference purchase order. If neither line item nor expense is

Case	Details of Reference Purchase Order	Details of Inbound XML E-Document based from the Purchase Order	Details of Inbound E-Document Template	Expected Conversion Result
have an expense included.				included in the inbound e-document, conversion will not proceed.
The purchase order has item lines and no specified expense, but the inbound e-document has expense amounts	The purchase order has no specified expense amount and account, but has at least one line item.	The inbound XML e-document has expense amounts included, without corresponding accounts.	The e-document template is valid with correct mapping.	Conversion will only proceed if there is at least one line item included, and a Default Expense Account is specified in the vendor record. The expense amounts in the inbound e-document will be assigned to the Default Expense Account.

Converting Individual Inbound E-Documents into Vendor Bills

If the conditions and requirements for conversion have been met, you are ready to convert an inbound e-document into a vendor bill. See Prerequisites and Conditions for Conversion

To convert an inbound e-document into a vendor bill

1. Go to the E-Documents Portlet on the Home page, and then click the number under **Inbound E-Documents for Conversion**.
2. On the Inbound E-Document for Conversion results page, click the View link of the inbound e-document that you want to convert into a vendor bill.
3. On the Inbound E-Documents page, click **Convert**.

Note: Before proceeding with conversion, make sure that the date format in the XML e-documents is the same as the date format set in Set Preferences.

The progress of conversion is displayed as a banner on the page. If conversion is successful, the created vendor bill is displayed. On the **E-Document** subtab of the vendor bill, the **Inbound E-Document** field indicates the record link of the converted inbound e-document. The **E-Document Audit Trail** subtab of the inbound e-document record will indicate the details of the successful conversion and the status is set to Converted. Details of any error or failure in conversion will also be listed in the e-document audit trail. If the converted inbound e-document has an attached PDF file reference, the attachment can be viewed and downloaded on the **E-Document** subtab of the created vendor bill.

Converting an Inbound E-Document Without a Purchase Order Number

An inbound e-document without a purchase order number can be converted into a stand-alone vendor bill. Even if the inbound e-document is neither derived from, associated with, nor linked to an existing purchase order record, it can still be converted into a vendor bill. But the inbound e-document that will undergo conversion, must meet the following requirements:

- It must be a valid and well-formed XML document.

- It must include a reference number.
- It must include at least one item purchased or an expense.
- If it includes expense line, vendor record must have a Default Expense Account.

Convert this type of (stand-alone) inbound e-document by clicking **Convert** on the record. Or, use a script to run automatic bulk conversion of inbound e-document records with status For Conversion. For more information, see Deploying Automatic Bulk Conversion Script for Inbound E-Documents.

Converting Failed Inbound E-Documents

Inbound e-documents that initially failed conversion can be manually converted again, individually or in bulk, into vendor bills.

First, you must review the inbound e-documents that failed conversion and then resolve the errors in those e-documents. Without resolving the errors, the conversion process will fail again.

After resolving the errors, you need to search for the inbound e-documents that you fixed and finally proceed with the bulk conversion.

To search and convert inbound e-documents that failed initial conversion:

1. Go to the Electronic Documents portlet on the Home page, and then click the number under **Convert Failed Inbound E-Documents**.

   The Convert Inbound E-Documents page is displayed with search filters where you can define criteria for refining the search for failed inbound e-documents.
2. Enter a date in the **Date Created From** and **Date Created To** fields.

   Note: The Date Created From must be an earlier date than the Date Created To.

   The combined dates that you entered define a date range during which all inbound e-documents created within that period will be displayed as the result.
3. (Optional) Select a vendor in the **Vendor** field

   The Vendor field further refines your search criteria by searching for inbound e-documents from a particular vendor.

   Important: The search will not proceed if any inbound e-document within the specified date range or from the selected vendor, is undergoing the conversion process at the same time. In this case, a message is displayed instructing you to change your search criteria or try again later when the ongoing conversion is finished.
4. Click **Search**.

   A results page is displayed listing the failed inbound e-documents that meet your search criteria. The results page displays the first 25 inbound e-documents, if there are more inbound e-documents found, they are displayed on succeeding pages. On the results page, you can click the Internal ID link of an inbound e-document is a link that to open the inbound e-document record.
5. Click **Convert**.

   A message is displayed on the banner, confirming that the e-documents are being converted.

After the conversion is completed, an email notification is sent to the user who performed the conversion and to the email address in the Email Recipient Notification setup, if any is specified. If errors are encountered during conversion, the email will have an attached CSV file listing the inbound e-documents that underwent conversion, including columns for the Internal ID, Vendor, and details of the conversion process and errors.

For more information about choosing the recipient of notifications, see step 4 of Prerequisites for Using Electronic Invoicing.

Note: You can also convert inbound e-documents that failed conversion, individually or one at a time. For more information, see Converting Individual Inbound E-Documents into Vendor Bills.

Compatibility of Approval Workflows with Vendor Bill Conversion

If Vendor Bill Approval Workflow is enabled, it is automatically applied to vendor bills created from inbound e-document conversion, without further configuration.

With approval workflow enabled, a vendor bill created from inbound e-document conversion is also assigned an approval status. If the new vendor bill is still pending approval when it was created, the approval process is continued. The vendor bill is then properly routed to the next approver defined in the workflow. For more information, see the help topic Vendor Bill Approval Workflow.

If 3 Way Match Vendor Bill Approval Workflow is enabled, you can determine the discrepancy between the created vendor bill and the reference purchase order. Clicking Bill Exception on the new vendor bill will display any discrepancy resulting from the validation performed by the approval workflow against set exception criteria. A vendor bill and its corresponding purchase order may have discrepancies in terms, quantity tolerance, quantity difference, or amount. For more information, see the help topic 3 Way Match Vendor Bill Approval Workflow.

Note: The standard workflow included in the Vendor Approval SuiteApp does not support checking for tax amount discrepancy. If you want to enable checking for tax amount discrepancy, see the help topic Customizing for the Vendor Bill Approval Workflow.

Canceling Inbound E-Documents

You can cancel an inbound e-document if you do not want to convert it into a NetSuite transaction.

To cancel an inbound e-document, view or open the inbound e-document you want to cancel and then click **Cancel**. A banner is displayed on the inbound e-document confirming cancellation of the record. The E-Document Audit Trail on the E-Document subtab also indicates the details of cancellation. Canceled e-documents cannot be converted to NetSuite records anymore.

You cannot cancel an inbound e-document if it is already converted or canceled. A canceled or converted e-document does not display the Cancel button. Cancellation will also fail if the inbound e-document is concurrently being converted or canceled by another user.

If cancellation of the record failed, a banner is displayed on the inbound e-document indicating the failure. The E-Document Audit Trail on the E-Document subtab also indicates the cause of the failure to cancel the inbound e-document.

Electronic Invoicing Errors

To understand outbound and inbound e-document processing errors, see the following topics:

- Electronic Invoicing Error Codes
- Outbound E-Document Generation Errors
- Outbound E-Document Sending Errors
- Inbound E-Document Conversion Errors

Electronic Invoicing Error Codes

Error code	Message	Description	Solution
EI_TEMPLATE_CSV_ERROR	The XML template contains errors. XML format must be well-formed.	The XML input for Template for Outbound E-Invoices field is invalid.	You must fix the XML input for the **Template for Outbound E-Invoices** field.
EI_TEMPLATE_VALIDATOR_CSV_ERROR	The **REGEX** field contains an incorrect regular expression. Proper syntax must be used.	The Regex input for XML validators is invalid.	You must fix the regex input for **Regex Validation Expression** field in the XML Validators sublist of the template.
EI_INACTIVE_CUSTOMER	Transactions with inactive customers are not supported by e-document.	The customer who owns the transaction is inactive.	You must remove the e-document template from the transaction or activate the inactive customer.
EI_SENDING_NO_RECIPIENTS	The e-document cannot be sent because the customer has no email address. Before you can send this e-document by email, an email address must be provided on the customer record.	The customer (individual) does not have any email address specified in the Customer record.	You must enter a valid email address in the Customer record.
EI_SENDING_NO_RECIPIENTS	There are no e-document recipients for this customer. To send electronic documents by email to this customer, at least one contact must be added to the list of e-document recipients.	The customer (company) does not have any e-document recipients defined in the Customer record.	You must enter e-document recipients in the Customer record.
EI_SENDING_RECIPIENT_NO_EMAIL	One or more recipients of the e-document, associated with this transaction, does not have an email address. Verify that the recipients for this customer have valid email addresses.	The customer (company) has an e-document recipient that does not have an email address.	You must enter a valid email address in the e-document recipient in the Customer record.
EI_SENDING_INVALID_METHOD	Select a valid sending method for {TYPE} #{TRANSACTIONNUMBER}.	The transaction does not have an e-document sending method defined.	You must enter an e-document sending method in the Transaction record.
EI_SEND_INVALID_RESULT	The plug-in failed to return a valid result.	The plug-in used for sending, did not return a result.	You must fix the plug-in implementation so that it will return a result object.
EI_SENDING_PLUG_IN_ERROR	<ERROR MESSAGE>	The plug-in used for sending, encountered an error.	You must fix the plug-in implementation based on the error message.
PROBLEM_LOADING_PLUG_IN	E-document Sending encountered an error while loading a custom plug-in. Error Code: <ERROR CODE> Message: <MESSAGE>	The plug-in file used for sending, did not load properly.	You must fix the plug-in implementation based on the error message.
EI_CANNOT_CREATE_DEFAULT_DOCUMENT_PACKAGE	The {DEFAULT_DOCUMENT_PACKAGE} record already exists. You cannot create an e-document package record with the same name. Rename your e-document package record and try again.	Creating an e-document package with the name "Default E-Document Package" is not allowed. (this error is shown in CSV import).	You must create a new e-document package with another name.

Error code	Message	Description	Solution
EI_CANNOT_EDIT_DEFAULT_DOCUMENT_PACKAGE	Editing the {DEFAULT_DOCUMENT_PACKAGE} record Name or Description is not allowed.	Editing the e-document package, "Default E-Document Package", is not allowed. (this error is shown in CSV import).	Avoid editing the default e-document package record.
EI_CANNOT_DELETE_DEFAULT_DOCUMENT_PACKAGE	Deleting the {DEFAULT_DOCUMENT_PACKAGE} record is not allowed.	Deleting the e-document package, "Default E-Document Package", is not allowed. (this error is shown in CSV import)	Avoid deleting the default e-document package record.
EI_VENDOR_CODE_FIELD_NOT_FOUND	The vendorcode field is missing in the e-document template. Modify the e-document template or select another template that includes vendorcode field mapping.	If the Multiple Vendor feature is enabled in the account, the vendorcode must be indicated in the Mapping Format of the E-Document Template that is being used for Inbound E-Document conversion.	You must add the mapping for vendorcode field in the Field Mapping for Inbound E-Documents in the template.
EI_NO_VENDOR_CODE_VALUE	At least one of the items has no vendor code. Cancel this e-document and submit another e-document with the correct value for the XML element mapped to the vendor code field.	The vendorcode of one of the items has no value. This error occurs if the Multiple Vendor feature is enabled in the account.	You must modify the XML by defining a value for the XML that is mapped to the vendorcode field.
EI_VENDOR_NAME_FIELD_NOT_FOUND	The vendorname field is missing in the e-document template. Modify the e-document template or select another template that includes vendorname field mapping.	If the Multiple Vendor feature is disabled in the account, the vendorname field must be indicated in the Mapping Format of the E-Document Template that is being used for Inbound E-Document conversion.	You must add the mapping for the vendorname field in the Field Mapping for Inbound E-Documents in the template.
EI_NO_VENDOR_NAME_VALUE	At least one of the items has no vendor name/code. Cancel this e-document and submit another e-document with the correct value for the XML element mapped to the vendor name/code field.	The vendorname of one of the items has no value. This error occurs if the Multiple Vendor feature is disabled in the account.	You must modify the XML by defining a value to the XML that is mapped to the vendorname field.
EI_CREATED_FROM_TRANSACTION_NOT_FOUND	Record ({TRANSTYPE}#{TRANSID}) was not found in the system. Cancel this e-document and submit another e-document with the correct value for the XML element mapped to the createdfrom field.	The transaction that the vendor bill will be created from is not found in the system.	You must define the correct value for the XML that is mapped to the createdfrom field.
EI_CREATED_FROM_TRANSACTION_ENTITY_NOT_MATCH	Record ({TRANSTYPE}#{TRANSID}) is assigned to a different entity. Select the correct entity and convert this e-document.	The referenced transaction has a different entity compared to the inbound e-document.	You must select the correct entity and convert the e-document again.
EI_TRANSACTION_REF_NUM_ALREADY_EXISTS	A vendor bill with the same reference number already exists. Cancel this e-document and submit another e-document with the correct reference number value for the XML element mapped to the tranid field.	This error occurs during conversion of inbound e-document to vendor bill. The SuiteApp will not proceed with conversion if the reference number on the inbound e-document already exists in one of the vendor bills.	You must define another value for the XML that is mapped to the tranid field.

Error code	Message	Description	Solution
EI_TRANSACTION_REF_NUM_NOT_FOUND	The required reference number is missing in the inbound e-document. Cancel this e-document and submit another e-document that includes an XML element for the reference number, mapped to the tranid field.	Reference number must be present in the XML. It has to be mapped to the E-Document template tranid field.	You must add an XML that maps to the tranid field.
EI_TRANSACTION_ITEM_ERROR	The following vendor codes: {ITEMLIST}, are associated with multiple item records. Modify the item records and ensure that vendor codes are unique for each item per vendor.	There are items in the XML (in the inbound e-Document) that have multiple matches in the item record.	You must edit the vendor codes of the duplicate item records to ensure the uniqueness of the items.
EI_TRANSACTION_ITEM_ERROR	The following vendor name/codes: {ITEMLIST}, are associated with multiple item records. Modify the item records and ensure that vendor name/codes are unique for each item per vendor.	There are items in the XML (in the inbound e-document) that have multiple matches in the item record.	You must edit the vendor name/codes of the duplicate item records to ensure the uniqueness of the items.
EI_TRANSACTION_ITEM_ERROR	The following vendor codes: {ITEMLIST}, are not associated with any item records.	There are items in the XML (in the inbound e-document) that have no match in the system.	You must create an item record for those items with no matches.
EI_TRANSACTION_ITEM_ERROR	The following vendor name/codes: {ITEMLIST}, are not associated with any item records.	There are items in the XML (in the inbound e-document) that have no match in the system.	You must create an item record for those items with no matches.
EI_TRANSACTION_ITEM_ERROR	There are no items in the inbound e-document that is included in the referenced transaction. Check the status of the referenced transaction if it can be transformed. If it can be transformed, cancel this e-document and submit another e-document with the correct value for the XML element mapped to the createdfrom field.	The user cannot convert the transaction or the referenced transaction does not have any matching items to the items in the inbound e-document.	A different transaction must be referenced or a different inbound e-document with the correct items must be used.
EI_SENDING_IN_PROGRESS	The system cannot perform a search using the filters you selected because e-document sending is already in progress for transactions within the date range ({TRANDATE_FROM} - {TRANDATE_TO}) for subsidiary ({SUBSIDIARY}). Please change your search criteria or try again later.	The records you are searching for might already be undergoing the sending process.	You can change the parameters for sending, or wait for the current sending process to finish before sending again.
EI_CONVERSION_IN_PROGRESS	The system cannot perform a search using the filters you selected because inbound e-document conversion is already in progress within the date range ({DATECREATED_FROM} - {DATECREATED_TO}). Please change your search criteria or try again later.	The records you are searching for might already be undergoing the conversion process.	You can change the parameters for conversion, or wait for the current conversion process to finish before converting again.
EI_CERTIFICATION_SENDING_METHOD_ALREADY_EXISTS	You cannot set this sending method as the Certification Sending Method because '{CERTIFICATION_SENDING_METHOD_NAME}' is already selected as the Certification Sending Method for the Subsidiaries [{SUBSIDIARIES}] and transactions [{TRANSACTIONS}]. To save this record, you must unassign '{CERTIFICATION_SENDING_METHOD_	A certification sending method is assigned to a combination of a transaction and subsidiary that already has a certification sending method. Only one certification sending method can be assigned to	You must unassign the existing certification sending method or make it inactive, to be able to assign a new certification sending method and save it.

Error code	Message	Description	Solution
	NAME}' as a Certification Sending Method, or make it inactive.	a particular combination of transaction and subsidiary.	
EI_TEMPLATE_ERROR	Any of the following error messages can be displayed: ■ You selected an outbound transaction type, but the template content in the **Template for Outbound E-Documents** field is missing. Enter the XML or JSON content in the field and try again. ■ There are missing field values. For an outbound transaction, specify a valid XML or JSON content in the **Template for Outbound E-Documents** field. For an inbound transaction, specify the JSON content in the **Field Mapping for Inbound E-Documents** field. ■ The selected XSD file is not a valid XSD file. Ensure that the file you select has the .xsd extension.	Any of these errors will be displayed if validations related to template are not met. For example, if any of the required fields **Template for Outbound E-Documents** or **Field Mapping for Inbound E-Documents** are missing, or when an invalid XSD file is selected.	You must provide the missing fields or select a valid XSD file.

Electronic Invoicing Common Errors

Message	Error on User Interface	Description	Solution
An error occurred during conversion. Check the E-Document Audit Trail on the E-Document subtab for details.	Banner on the record	A red banner is displayed on the inbound e-document record indicating failed conversion.	Check the audit trail for more information.
Unable to convert this inbound e-document because the selected vendor is inactive. The E-Document Status field has not been updated and an audit trail has not been created. Clear the Inactive box on the vendor record, then try converting the e-document again.	Banner on the record	A yellow banner is displayed on the inbound e-document record indicating that an inactive vendor is selected.	Activate the inactive vendor before converting the inbound e-document.
Unable to convert this inbound e-document because the selected customer is inactive. The E-Document Status field has not been updated and an audit trail has not been created. Clear the Inactive box on the customer record, then try converting the e-document again.	Banner on the record	A yellow banner is displayed on the inbound e-document record indicating that an inactive customer is selected.	Activate the inactive customer before converting the inbound e-document.
Parsing failure. Check the Field Mapping for Inbound E-documents.	Audit Trail log	The Audit Trail of the inbound e-document logs this error after a failed conversion due to failure in XML parsing.	Check the inbound template for errors.
Conversion failure.	Audit Trail log	The Audit Trail of the inbound e-document logs	Check the inbound e-document for errors.

Message	Error on User Interface	Description	Solution
		the error after the failed conversion process.	
Conversion failed because the status of the inbound e-document record is '{STATUS}'	Audit Trail log	The Audit Trail of the inbound e-document logs the error after a failed conversion due to a wrong status detected during conversion process.	A concurrent conversion process might be running, or the record is already cancelled when conversion was started. You can wait for the other conversion process to finish, or verify if the record has been canceled.
An error occurred during cancellation. Check the E-Document Audit Trail on the E-Document subtab for details.	Banner on the record	A banner is displayed indicating an error in cancellation.	You must check the inbound e-document for errors; most likely, the error is due to an invalid status.
The selected XML File Reference is not a valid XML file. Ensure that the file you select has the .xml extension.	Popup message	The inbound e-document record has an invalid e-document file.	You must make sure that the file attached to the inbound e-document is an XML file.
The selected XML File Reference is not a well-formed XML document.	Popup message	The inbound e-document record has a malformed XML file.	You must check the attached XML file if it is well-formed.
The Transaction Date From must not be later than the Transaction Date To. Change the dates so that the Transaction Date From is earlier than the Transaction Date To.	Popup message	The selected Transaction Date To is earlier than the Transaction Date From.	You must change the selected dates to make the Transaction Date From earlier than the Transaction Date To.
The Date Created From must not be later than the Date Created To. Change the dates so that the Date Created From is earlier than the Date Created To.	Popup message	The selected Date Created To is earlier than the Date Created From.	You must change the selected dates to make the Date Created From earlier than the Date Created To.
The selected XSD file is not a valid XSD file. Ensure that the file you select has the .xsd extension.	Popup message	This message is displayed while creating/editing a template record when the selected file is not an XSD file.	You must select an XSD file
There is no e-document email sender for this vendor. To receive e-documents through email from this vendor, you must enter at least one email address in the Vendor E-Document Email Sender list.	Popup message	The vendor does not have any e-document senders defined in the vendor record. This error appears when saving the record with Use Sender List box checked.	You must add at least one sender email address, or clear the box and input a sender domain.
The sender email address already exists.	Popup message	The sender email address entered in the sublist is already existing for the same vendor.	You must enter a different email address or remove the current entry.
The sender email domain is already being used by a different vendor.	Popup message	The sender domain is already being used by another vendor.	You must enter a different domain, or use the sender list to enter specific email addresses.

Message	Error on User Interface	Description	Solution
This account does not have an active license to use the Electronic Invoicing SuiteApp in multiple countries. To convert e-documents in bulk, please contact your account administrator to configure the E-Document Country for Free Use on the Company Information page.	Popup message	This message is displayed when trying to convert failed inbound e-documents in bulk using the Suitelet.	The administrator must set up the E-Document Country for Free Use field.
This account does not have an active license to use the Electronic Invoicing SuiteApp in multiple countries. To convert this e-document to a transaction, please contact your account administrator to specify a country in the E-Document Country for Free Use field on the Company Information page.	Popup message	This message is displayed when trying to convert an inbound e-document individually from the record page.	The administrator must set up the E-Document Country for Free Use field.
This account does not have an active license to use the Electronic Invoicing SuiteApp in multiple countries. To convert this e-document to a transaction, please contact your NetSuite account manager to purchase a license.	Popup message	This message is displayed when trying to convert an inbound e-document individually from the record page.	Ask the account manager to purchase a license.
This account does not have an active license to use the Electronic Invoicing SuiteApp in multiple countries. To convert this e-document to a transaction, please set up the default billing address of the selected vendor.	Popup message	This message is displayed when trying to convert an inbound e-document individually from the record page.	You must add a default billing address to the vendor of this e-document.

Outbound E-Document Generation Errors

The following errors can occur during the generation of e-documents:

- Malformed Template
- Malformed XPath
- Malformed Regex
- Unsupported XPath
- Data Failed Validation Checking
- Inactive Customer Record

Malformed Template

Generation of e-documents will fail if the selected template contains FreeMarker syntax errors.

To fix this error, select a different template or update the content of the e-document template, then try generating the e-document again.

Make sure that e-document templates are well-formed.

Click the link to the e-document template in the **E-Document Template** field to view and edit your e-document template.

For more information, see XPath and Regex Examples for E-Document Templates.

Malformed XPath

Generation of e-documents will fail if there is no xpath to validate your e-document template, or the xpath format is invalid.

To fix this error, select a different template or verify that an xpath exists and the format is correct in your e-document template, then try generating the e-document again.

Make sure that e-document templates use the proper syntax for the xpath.

Click the link to the e-document template in the **E-Document Template** field to view and edit your e-document template.

For more information, see XPath and Regex Examples for E-Document Templates.

Malformed Regex

Generation of e-documents will fail if the regex validation expression that validates your e-document template is invalid.

To fix this error, select a different template or check the regex validation expression's format in your e-document template, then try generating the e-document again.

Make sure that e-document templates use the proper syntax for the regex validation expression.

Click the link to the e-document template in the **E-Document Template** field to view and edit your e-document template.

For more information, see XPath and Regex Examples for E-Document Templates.

Unsupported XPath

Generation of e-documents will fail if the e-document template does not contain the information required by the xpath used to validate it.

To fix this error, select a different template, update the e-document template, or delete the xpath, then try generating the e-document again.

Click the link to the e-document template in the **E-Document Template** field to view and edit your e-document template.

For more information, see XPath and Regex Examples for E-Document Templates.

Data Failed Validation Checking

Generation of e-documents will fail if the transaction does not contain the information needed by the e-document template you selected. For example, if your e-document template requires a shipping address with a 5-digit postal code, but the shipping address on your transaction record has a 4-digit postal code, an error will occur.

To fix this error, select a different template, update the transaction record with the required information, or update the e-document template validations, then try generating the e-document again.

Inactive Customer Record

Generation of e-documents will fail if the customer associated with the transaction record is inactive.

To fix this error, set the **E-Document Template** field on the transaction to blank. If the **E-Document Template** field is blank, the system will not generate an e-document for the transaction, and no error message will be shown. If you intend to generate an e-document for the transaction, you must make the customer record active, then try generating the e-document again.

To activate the customer record, go to Lists > Relationships > Customers and find the customer record. Click **Edit**, then go to the **System Information** subtab and clear the **Inactive** box.

To regenerate e-documents, see the following topics:

- Regenerating E-Documents for Single Transactions
- Generating and Regenerating E-Documents in Bulk

Outbound E-Document Sending Errors

Errors can occur when sending e-documents. When the system encounters an error, NetSuite sends an email notification containing the error details to the user who initiated the sending process.

An audit trail containing the error details is also created on the **E-Document Audit Trail** subtab on the transaction record.

Fixing errors may require an Administrator role. Be sure to inform your account administrator about the error immediately. If the error is not fixed, the e-document cannot be sent.

The following errors can occur when sending e-documents:

- Invalid Sending Method
- Invalid E-Document Sender
- Invalid Email Recipient
- Plug-in Script Error

Invalid Sending Method

Sending of e-documents will fail if no sending method has been selected on the transaction record.

To fix this error, edit the transaction record and select a sending method, then try sending the e-document again.

Invalid E-Document Sender

Sending of e-documents will fail if the e-document sender has no email address.

To fix this error, change the e-document sender or make sure the e-document sender has a valid email address, then try sending the e-document again.

To view the **E-Document Sender** field, go to Setup > Company > Company Information.

If you have a OneWorld account, go to Setup > Company > Classifications > Subsidiaries. Then, click the subsidiary to view the assigned employee in the **E-Document Sender** field.

To add or change the email address, go to Lists > Employees and edit the employee record.

Invalid Email Recipient

Sending of e-documents will fail if the selected sending method has no email recipients or if one or more email recipients have no email addresses.

To fix this error, select a different sending method or make sure the selected sending method has at least one email recipient with an email address. If the customer is a company, there should be at least one email recipient defined on the **E-Document Email Recipient** subtab on the customer record. If the customer is an individual, enter the email address.

To view and add email recipients for the selected sending method, go to the **E-Document** subtab of the customer record. On the **E-Document Email Recipient** field, verify that your email recipients have email addresses. Click **New E-Document Email Recipient** to add an email recipient. Only contacts associated with the customer record are available for selection in the dropdown list.

Plug-in Script Error

Sending of e-documents will fail if the selected custom sending method is invalid.

To fix this error, make sure that the script of your custom sending method is valid.

For more information, see Creating Custom Methods for Sending E-Documents.

Inbound E-Document Conversion Errors

Errors can occur when converting e-documents. When an error is encountered in automatic scheduled bulk conversion, the system sends an email notification with error details to the Recipient of E-Document Notifications. If the Recipient of E-Document Notifications is not assigned, the system sends the notification to all active administrators.

An audit trail containing the error details is also created on the **E-Document Audit Trail** subtab on the inbound e-document record.

Fixing errors may require an Administrator role. Be sure to inform your account administrator about the error immediately. If the error is not fixed, the e-document cannot be converted.

The following errors can occur when converting inbound e-documents:

- Purchase Order is not Ready
- Incorrect PO Number
- Item vendorcode is not Unique
- Duplicate Reference Number
- Incorrect Mapping to Transaction Fields
- Mapping to Required Vendor Bill Fields is Missing
- Missing Default Expense Account

Purchase Order is not Ready

Inbound e-document conversion will fail, if the e-document to be converted was generated from a purchase order that is not ready for billing, closed or fully billed.

To fix this error, ensure that the PO status is ready for billing.

Incorrect PO Number

Inbound e-document conversion will fail, if the e-document to be converted was generated from a purchase order with incorrect PO Number.

To fix this error, contact the vendor or party whom the e-document came from, inform them to make corrections to the PO Number, and have them send the amended XML file.

Item vendorcode is not Unique

Inbound e-document conversion will fail, if the e-document to be converted contains items whose vendorcode are not unique.

To fix this error, update the item records and ensure that each item has a unique vendorcode.

Duplicate Reference Number

Inbound e-document conversion will fail, if the Reference Number of an e-document to be converted has a duplicate. Duplicate Vendor Bill Detection is enabled for the Electronic Invoicing SuiteApp, so a duplicate Reference Number and vendor will cause the error.

To fix this error, review the existing bill, and then determine if it is duplicate. If it is a duplicate, cancel the e-document. If it is not duplicate, ask the vendor to resend an updated XML document, and then cancel the e-document.

Incorrect Mapping to Transaction Fields

Inbound e-document conversion will fail, if the JSON template has incorrect mapping to transaction fields.

To fix this error, review the inbound e-document template and make sure that JSON objects map to the correct vendor bill transaction fields.

Mapping to Required Vendor Bill Fields is Missing

Inbound e-document conversion will fail, if the JSON template is missing the mapping to required vendor bill fields.

To fix this error, review the inbound e-document template and define the JSON objects that will map to required vendor bill transaction fields.

Missing Default Expense Account

Inbound e-document conversion will fail, if the XML contains an expense line but the vendor record does not have a default expense account specified. To fix this error, edit the vendor record and on the **Financial** tab, specify a Default Expense Account.