

# Perceptive Accounts Payable Invoice eForm

Supplemental Guide

Version: 10.5.x

Compatible with ImageNow Versions: 6.5.1.149 to 6.7.x

The logo consists of a horizontal bar with a color gradient from blue to green, ending in a right-pointing arrow shape.

**perceptive**software

Written by: Product Documentation, R&D  
Date: June 2013

© 2013 Perceptive Software. All rights reserved

CaptureNow, ImageNow, Interact, and WebNow are trademarks of Lexmark International Technology SA, registered in the U.S. and other countries. Perceptive Software is a stand-alone business unit within Lexmark International Technology SA. All other brands and product names mentioned in this document are trademarks or registered trademarks of their respective owners. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or any other media embodiments now known or hereafter to become known, without the prior written permission of Perceptive Software.

# Table of Contents

- Compatibility matrix .....7**
- Virtual tables .....7**
  - Required files..... 7
    - Business Unit*..... 7
    - GL Account*..... 8
    - GL Code*..... 9
    - Location* ..... 9
    - Purchase Orders*..... 10
    - Purchase Order Lines* ..... 10
    - Vendor*..... 12
  - Optional files..... 13
    - AOC GL Account* ..... 13
    - AOC GL Code*..... 14
    - Country Currency*..... 15
    - Currency Codes*..... 16
    - GL User*..... 16
    - Payment Terms*..... 17
    - Special Handling*..... 17
    - SUT Codes*..... 18
    - SUT Apply Codes* ..... 18
    - VAT Code*..... 18
    - Vendor Location*..... 19
    - VAT Registration Number (VAT ID) Vendor* ..... 19
    - VAT Registration Number (VAT ID) Business Unit*..... 20
  - Virtual Table Updater configuration file ..... 21
    - AP\_VirtualTable\_Updater.xml*..... 21
  - Virtual tables updates ..... 22
  - Archive processed files ..... 22
- Document types and custom properties .....23**
  - Invoice ..... 23
  - Credit Memo..... 24
- AP Invoice eForm file directories .....25**
  - \inservice6\etc\ap* ..... 25

\inserver6\temp\ap .....	26
\inserver6\script.....	27
\inserver6\form\[ <i>build folder</i> ] .....	29
\inserver6\form\data_definition .....	30
<b>AP Invoice eForm log files .....</b>	<b>30</b>
<b>AP Invoice eForm configuration files.....</b>	<b>31</b>
AP_Skin.xml .....	31
AP_Config.xml .....	40
AP_Export.xml .....	55
AP_Validate_Server.xml .....	62
AP_Custom_Cultures.xml .....	69
AP_VirtualTable_Updater.xml.....	71
<b>Configure workflow for AP Invoice eForm .....</b>	<b>72</b>
About the reindex document and custom property mapping script .....	72
About the validation script .....	72
About workflow queue roles.....	73
Reserved characters in XML and forms .....	74
<b>Custom Logic.....</b>	<b>75</b>
Custom functions .....	75
Global variables.....	75
Add custom business logic to the form .....	76
JavaScript event handlers .....	76
Call a server-side iScript .....	77
Enhance the validation iScript.....	78
Enhance the export iScript.....	79
<i>Field level sample .....</i>	<i>79</i>
<i>Global level sample.....</i>	<i>79</i>
<b>Demonstration setup of AP Invoice eForm.....</b>	<b>80</b>
Load Data .....	80
Create AP workflow process.....	81
Configure iScript files and routing actions .....	82
Create an Invoice Entry Validation routing rule .....	82
<i>Create a rule to route back for header error conditions.....</i>	<i>82</i>
<i>Create a routing rule for non-PO invoices.....</i>	<i>83</i>

<i>Create a routing rule for PO-based invoices within tolerance .....</i>	84
<i>Create a routing rule for PO-based invoices outside of tolerance .....</i>	84
Create a GL Coding Validation routing rule .....	85
Configure workflow queue properties .....	86
<b>Configure validation within the eForm .....</b>	<b>87</b>
Enable validation for specific roles and queues .....	87
Configure the duplicate invoice check .....	87
<i>Duplicate invoice check .....</i>	88
<i>Advanced duplicate invoice check .....</i>	88
Configure header-level and line-level required fields .....	89
Configure GL code validation .....	90
<i>Configure the GL Codes source values .....</i>	90
<i>Configure the GL User and GL Code virtual tables with constraints .....</i>	91
<i>Enable the GL code validation .....</i>	91
Configure GL account validation .....	92
Configure the Invoice Total field validation .....	92
<i>Configure amount validation and distribution in PO invoices .....</i>	92
<i>Configure Invoice Total validation in non-PO invoices or credit memos .....</i>	94
PO/invoice currency validation .....	95
VAT validation .....	95
<b>Configure value added tax (VAT) within the eForm .....</b>	<b>96</b>
Create and enable the VAT codes .....	96
<i>Create or update the virtual tables .....</i>	96
<i>Update AP_VirtualTable_Updater.xml .....</i>	96
<i>Configure AP_Skin.xml .....</i>	96
Enable the VAT amount summary .....	97
Enable VAT amount validation .....	97
<i>Update the Currency Codes virtual table .....</i>	97
<i>Update AP_VirtualTable_Updater.xml .....</i>	97
<i>Configure AP_Skin.xml .....</i>	98
<i>Configure AP_Validate_Server.xml .....</i>	98
Associate the vendor VAT registration numbers .....	98
<i>Create the VAT Registration Number (VAT ID) Vendor virtual table .....</i>	98
<i>Configure AP_Skin.xml .....</i>	99

- Associate the sold-to VAT Registration Numbers ..... 99
  - Create the VAT Registration Number (VAT ID) Business Unit virtual table ..... 99
  - Configure AP\_Skin.xml ..... 99
- Enable the VAT Compliance check box ..... 99
- Configure add on cost (AOC) within the eForm ..... 100**
  - Create and enable the AOC codes ..... 100
    - Create the AOC virtual tables ..... 100
    - Update AP\_VirtualTable\_Updater.xml ..... 100
    - Enable GL distribution with AOC codes in AP\_Skin.xml ..... 101
  - Enable multi-line GL distribution exports for Lawson ..... 102
    - Enable PO GL distributions in AP\_Config.xml ..... 102
    - Configure the miscellaneous line export in AP\_Export.xml ..... 102
- Globalization ..... 103**
  - Supported languages, countries and currencies ..... 103
  - Additional supported currencies ..... 105
  - Configuration considerations ..... 105
    - Windows regional settings ..... 105
    - Keyboard configuration ..... 105
    - Language Bar ..... 105
- Appendix A: Add a custom currency example ..... 106**
- Appendix B: User interface labels, tooltips, and messages ..... 107**
  - User interface labels ..... 107
  - Tooltips ..... 115
  - Error and confirmation messages ..... 116
- Index ..... 119**

## Compatibility matrix

The following table maps the compatible versions of ImageNow Server and Perceptive Accounts Payable Invoice eForm.

For versions of Accounts Payable Invoice eForm prior to 10.0.0.28, contact your Perceptive Software representative to determine compatibility.

ImageNow Server	Perceptive AP Invoice eForm
6.5.1.149 to 6.6.x	10.0.0.28 to 10.5.x
6.7.x	10.1.0.307 to 10.5.x

## Virtual tables

To load the virtual tables, you create several Comma Separated Value (CSV) files. Each file has a different format. Do not change the TableName column for any of these files.

### Required files

#### Business Unit

The Bus\_Unit table contains an entry for every Business Unit associated with AP invoices. Many business applications have a VendorGroup value. If none is available, use the letter A as the value.

#### Value added tax (VAT)

For solutions implementing value added tax (VAT): to provide a default Jurisdiction or VAT registration number (VAT ID) value for each business unit, populate the Jurisdiction and VAT ID fields. If left blank, the user can manually key the Jurisdiction and VAT ID values into the eForm, as necessary. For an overview of incorporating VAT into the eForm, refer to the Configure value added tax (VAT) within the eForm section of this document.

	A	B	C	D	E	F	J
1	Table Name*	Active*	Bus Unit ID*	Bus Unit Name*	Vendor Group*	Jurisdiction	VAT ID
2	BUS_UNIT	Y	1234	ACME Construction	ACME	US	
3	BUS_UNIT	Y	4500	CGI Corporation	CGI	FR	FRXX999999999
4	BUS_UNIT	Y	9001	International Retail	2	FR	FR1234567890
5	BUS_UNIT	Y	4321	LGE Corporation	4321	DE	DE123456789
6	BUS_UNIT	Y	9000	UK Retail Ltd	1	GB	GB999 9999 73

\*Indicates a required column

## GL Account

This table contains an entry for every general ledger (GL) account code combination, for use during invoice coding.

There is a maximum of six elements to the GL Account Code (columns C, D, E, F, H, and I). In the event that you use fewer elements, you can redefine the table by setting the default value to a null value. The GL Account columns 5-8 are optional, and provide additional flexibility. If you do not use these elements, update the default value to 0 (zero).

	A	B	C	D	E	F
1	Table Name*	Active*	Bus Unit ID*	Accounting Unit*	GL Account*	GL Sub Account*
2	GL_ACCT	Y	1234	101	17900	100
3	GL_ACCT	Y	1234	101	17900	200
4	GL_ACCT	Y	1234	101	17900	300
5	GL_ACCT	Y	1234	101	22500	100
6	GL_ACCT	Y	1234	101	22500	200
7	GL_ACCT	Y	1234	101	22500	100

	G	H	I	J	K
1	GL Account Desc	GL Account 5	GL Account 6	GL Account 7	GL Account 8
2	Office Supplies - Misc.	100	100	0	0
3	Office Supplies - Copiers	100	100	0	0
4	Office Supplies - Printers	200	200	100	1008
5	Maintenance – Buildings	200	200	100	1008
6	Maintenance – Grounds	200	200	100	1008
7	Maintenance – HVAC	0	0	200	2008

\*Indicates a required column

## GL Code

The GLCODE virtual table includes an entry for every general ledger code. Each combination of Constraint1, Constraint2, GLNumber and GLCode must be unique. The GL User virtual table defines which GL Codes each user can access.

To provide varying levels of access to the GL code, you can set Constraint1, Constraint2, or both to ALL.

	A	B	C	D	E	F	G	
1	Table Name*	Constraint 1*	Constraint 2*	GL Number*	GL Code*	Description	Active*	
2	GLCODE	Department_01	KCMO	1	1101	Petty Cash	Y	
3	GLCODE	Department_01	KCMO	1	1102	Travel Advances	Y	
4	GLCODE	Department_05	KCMO	2	1103	Livestock	Y	
5	GLCODE	Department_21	KCMO	3	1104	Laptops	Y	

\*Indicates a required column

## Location

The Sales-Use-Tax-Code and Ultimate-Use-Tax-Code columns do not require default values. Each field designates the default value by location. These entries do not represent all the valid values for these fields. They are the default values to use for this location. The lists of all the valid values are in two other tables.

	A	B	C	D	D	E	G	
1	Table Name*	Set ID*	Location Code*	Location Name*	Sales-Use-Tax-Code	Ultimate-Use-Tax-Code	Active*	
2	LOCATION	SHARE	KC01	CARTER PLANT – KC MO	10100	A02	A	
3	LOCATION	SHARE	DET01	SMITH WAREHOUSE - DET	15600	B14	A	

\*Indicates a required column

## Purchase Orders

To use the AP Invoice eForm for PO-based invoices, you must create the Purchase Orders table.

The PO table contains an entry for every open Purchase Order. Use the Date Closed to control paid invoices. RemitTo is the Vendor Remit To Address Sequence Number.

	A	B	C	D	E	F		G	H	
1	Table Name*	Bus Unit ID*	PO Number*	Vendor ID*	PO Amount*	PO Currency	PO Date	Date Closed*	Remit To*	
2	PO	1234	650000109	3004	127.3	USD	7/2/2008		2	
3	PO	1234	650000110	3004	158.17	USD	7/8/2009		1	
4	PO	1234	650000201	3004	584.17	EUR	8/1/2009		1	
5	PO	1234	650001001	1001	7300.8	GBP	7/20/2009		1	
6	PO	1234	650001012	1001	538.8	EUR	8/1/2009		1	
7	PO	1234	650000920	2000	1150	CAD	8/2/2007		1	

\*Indicates a required column

## Purchase Order Lines

To use the AP Invoice eForm for PO-based invoices, you must create the Purchase Order Lines table.

The PO\_LINE table contains a line item entry for every Purchase Order loaded into the PO Table. The table requires the following columns.

If using VAT fields in your eForm, use the Jurisdiction and VATCode columns. For an overview of incorporating VAT into the eForm, refer to the [Configure value added tax \(VAT\) within the eForm](#) section of this document.

QTY received less the QTY invoiced (or matched). Verify this calculation based on your purchase order system and the data you store within the PO line items. POItemQty might vary slightly based on the type of matching instituted. The POItemQty column reflects one of the following states, depending on the matching type:

- 2-way: quantity ordered
- 3-way: quantity received but not invoiced
- 4-way: quantity received in good condition but not invoiced

The POItemQty (PO Line Item Quantity) should reflect the Quantity Received, but not invoiced. In most situations, you may need to calculate this using business application data.

**Note** AP Invoice eForm supports negative amounts in the Purchase Order Details section.

	A	B	C	D	E	F	G
1	Table Name*	Bus Unit ID*	PO Number*	PO Line Number*	PO Item Number*	PO Item Desc*	PO Item Quantity*
2	PO_LINE	1234	650000109	1	B100	Binder Clips	20
3	PO_LINE	1234	650000110	2	CPW 16X	Copier Paper 200 Sheets	5
4	PO_LINE	1234	650000201	1	AAG-Desk	At-A-Glace Desk Pack	10
5	PO_LINE	1234	650001001	2	PPCAC	Plastic Paper Clips	20
6	PO_LINE	1234	650001012	3	TTD114	Things to Do Pads	5
7	PO_LINE	1234	650000920	1	1-PN	10 Penny Nails	10
	H	I	J	K	L	M	N
1	PO Item UOM*	PO Item Unit Price*	PO Item Ext Amt*	Taxable	Date Closed	Jurisdiction	VAT Code
2	BOX	2.49	49.8	Y		FR	V0
3	BOX	15.5	77.5	Y		FR	V1
4	BOX	6.39	63.9	N		FR	V2
5	BOX	3.19	63.8	N		FR	V3
6	BOX	3.89	19.45	Y		FR	V4
7	BOX	1.88	18.8	Y		FR	V5

\*Indicates a required column

## Vendor

The VENDOR table contains an entry for every active Vendor.

Most business applications have a Vendor Group value, but if none is available, use the letter A as the value. The Remit To value is an Address Sequence Number. The combination of Vendor ID and Remit To value comprises the actual Vendor number.

### Value added tax (VAT)

To provide a default VAT registration number (VAT ID) for each vendor, populate the VAT field. If left blank, the user can manually key a VAT registration number into the eForm. For an overview of incorporating VAT into the eForm, refer to the [Configure value added tax \(VAT\) within the eForm](#) section of this document.

	A	B	C	D	E	F	G	H
1	Table Name*	Active*	Vendor Group*	Vendor ID*	Vendor Name*	Address 1	Address 2	City
2	VENDOR	A	ACME	1000	GENERAL SUPPLIES	1600 MAIN		CHICAGO
3	VENDOR	A	ACME	1000	GENERAL SUPPLIES	2626 COMMERCE	Suite 100	NEW YORK
4	VENDOR	A	EUR	4500	Le Zèbre	2 avenue Gabriel		PARIS

	I	J	K	L	M	N	O	P	Q
1	State	Postal Code	Remit To*	Country Code	Vendor Short Name	Payment Terms	VAT ID	Field 1	Field 2
2	IL	60600	1	US	GENSUP	Net 30			
3	NY	86888	2	US	GENSUP	Net 30			
4		75008	1	FR	LZEBRE	Net 45	FR1X123456		

\*Indicates a required column

## Optional files

### AOC GL Account

The add on cost general ledger (AOC GL) account table contains an entry for each AOC GL account code combination, for use with distributed additional amounts. This table stores the AOC code and optional default values for the business unit, accounting unit, and up to six additional GL Account fields. When using AOC GL account codes in AP Invoice eForm, this table provides the default values for populating the GL Distributions section. In the eForm, the user may manually key in alternate values.

You can set the default AOC code in the AP\_Skin.xml file, for the Sales Tax, Freight, Misc, and Additional Amounts <1-8> fields. A default AOC code is not required.

	A	B	C	D	E	F	G
1	Table Name*	Active*	AOC Code*	AOC Desc.	Bus Unit ID*	Accounting Unit	GL Account
2	AOC_GL_ACCT	Y	FR	Freight	100	10100	17900
3	AOC_GL_ACCT	Y	FS	Fuel Surcharge	100	10100	22500
4	AOC_GL_ACCT	Y	HA	Handling	100	10100	22500
5	AOC_GL_ACCT	Y	IN	Insurance	100	10100	57020

	H	I	J	K	L	M
1	GL Sub Account	GL Account Desc	GL Account 5	GL Account 6	GL Account 7	GL Account 8
2	300	Freight, Express & Postage			200	
3	100	Fuel/Diesel Oil				2008
4	200	Miscellaneous	100		200	
5	100	Insurance	200			2008

\*Indicates a required column

## AOC GL Code

The add on cost general ledger (AOC GL) code table contains an entry for each AOC code and up to 24 optional GL Codes with an accompanying description. When using AOC GL codes in AP Invoice eForm, this table provides the default values for populating the GL Distributions section. In the eForm, the user may manually key in alternate values.

You can set the default AOC code in the AP\_Skin.xml file, for the Sales Tax, Freight, Misc, and Additional Amounts <1-8> fields. A default AOC code is not required.

	A	B	C	D	E	F
1	Table Name*	Active*	AOC Code*	AOC Desc	GL Code 1	GL Code Desc 1
2	AOC_GL_CODE	Y	FR	Freight	2000	Warehouse
3	AOC_GL_CODE	Y	FS	Fuel Surcharge	2000	Warehouse
4	AOC_GL_CODE	Y	HA	Handling	3000	Miscellaneous
5	AOC_GL_CODE	Y	IN	Insurance	3100	Insurance

	G	H	I	J	K	L
1	GL Code 2	GL Code Desc 2	GL Code 3	GL Code Desc 3	GL Code <4-24>	GL Code <4-24>
2	100	Freight, Express & Postage	110	Misc Expense	...	...
3	120	Fuel/Diesel Oil	130	Service	...	...
4	130	Miscellaneous			...	...
5	140	Insurance	110	Misc Expense	...	...

\*Indicates a required column

## Country Currency

The COUNTRYCURRENCY table supports the mapping of a country to a currency. This table is optional. If you do not populate the table, the eForm follows the number and percentage formatting rules of the originally presented eForm language.

### Notes

- The CountryCode value must match the CountryCode value defined in the VENDOR table.
- The CurrencyCode value must match the CurrencyCode value defined in the CURRENCY table.
- The LocaleCode must match one of the 2-character or 4-character language and country codes.

	A	C	B	D	E	
1	<b>Table Name*</b>	<b>Currency Code*</b>	<b>Country Code*</b>	<b>Locale Code*</b>	<b>Active*</b>	
2	COUNTRYCURRENCY	USD	US	en-US	Y	
3	COUNTRYCURRENCY	CAD	CA	en-CA	Y	
4	COUNTRYCURRENCY	EUR	DE	de-DE	Y	
5	COUNTRYCURRENCY	EUR	FR	fr-FR	Y	
6	COUNTRYCURRENCY	EUR	NL	nl-NL	Y	
7	COUNTRYCURRENCY	EUR	ES	es-ES	Y	
8	COUNTRYCURRENCY	EUR	IT	it-IT	Y	
9	COUNTRYCURRENCY	MXP	MX	es-MX	Y	
10	COUNTRYCURRENCY	BRL	BR	pt-BR	Y	
11	COUNTRYCURRENCY	GBP	GB	en-GB	Y	
12	COUNTRYCURRENCY	INR	IN	en-IN	Y	

\*Indicates a required column

## Currency Codes

The CURRENCY table contains an entry for all the Currency Codes by Business Unit.

For an overview of incorporating VAT into the eForm, refer to the [Configure value added tax \(VAT\) within the eForm](#) section of this document.

	A	B	C	D	E	F	G	H
1	Table Name*	Bus Unit ID*	Currency Code*	Currency Name*	Pay Group	Active*	Variance Amount	Variance Percent
2	CURRENCY	1234	USD	US Dollar	7000	A	1.33	
3	CURRENCY	1234	GBP	Pound	8100	A	.75	
4	CURRENCY	1234	EUR	Euro	7000	A		1.5
5	CURRENCY	1234	MXN	Peso	7000	A	12.5	

\*Indicates a required column

## GL User

The optional GLUSER table allows you to restrict which values each of your users can view on a field-by-field basis. To use this function, you must populate the GLUSER table with an entry for each general ledger user. You cannot use ImageNow groups. Note that you can use the authorization and restriction provided by this functionality on the ImageNow owner and administrator accounts. Each GL User has access to the GL Codes based on the constraints defined in Constraint 1 and Constraint 2, which correspond to constraints defined in the GL Code table.

Values you define for Constraint 1 and Constraint 2 must exist in the Constraint 1 and Constraint 2 columns in the GL Code table. If you leave the Constraint1 and Constraint2 columns blank, the user can only access GL Codes that have Constraint1 and Constraint2 values of ALL.

Use this virtual table only if you want to filter GL Code values by User. Otherwise, you do not need to use the GL User Table.

	A	B	C	D	E	
1	Table Name*	UserName*	Constraint 1*	Constraint 2*	Active*	
2	GLUSER	JDOE	Department_01	KCMO	Y	
3	GLUSER	JJONES	Department_05	KCMO	Y	

\*Indicates a required column

## Payment Terms

The PAYMENTTERMS table calculates a voucher payment due date based on the vendor's payment terms criteria and calculation method.

The eForm calculates the payment due date based on the formula you designate in PaymentCalcMethod. The available formulas include:

- **FROM\_INV\_DT** Payment Date = Invoice Date + Number of Days
- **FROM\_EOM** Payment Date = Last day of the month (based on invoice date) + Number of Days
- **DAY\_CURR** Payment Date = Day of the month [Number of Days value] for the current month (based on invoice date)
- **DAY\_NEXT** Payment Date = Day of the month [Number of Days value] for the following month (based on invoice date)

	A	B	C	D	E	
1	Table Name*	Payment Term Code*	Payment Calculation Method*	Payment Day Of Month*	Active*	
2	PAYMENTTERMS	Net 10	FROM_INV_DT	10	Y	
3	PAYMENTTERMS	Net 15	FROM_INV_DT	15	Y	
4	PAYMENTTERMS	Net 30	FROM_INV_DT	30	Y	

\*Indicates a required column

## Special Handling

The SPEC\_HANDLING table contains an entry for every Special Handling Code that the eForm process references.

	A	B	C	D	
1	Table Name*	Active*	Special Handling Code*	Special Handling Desc*	
2	SPEC_HANDLING	Y	EXP	Expedite Payment	
3	SPEC_HANDLING	Y	HOLD	Hold for Supplier Pickup	
4	SPEC_HANDLING	Y	ATT	Attach Supporting Documentation	

\*Indicates a required column

## SUT Codes

The SUT\_CODE table contains an entry for all the Sales Use Tax Codes.

	A	B	C	
1	<b>Table Name*</b>	<b>Active*</b>	<b>Tax Code*</b>	
2	SUT_CODE	Y	10100	
3	SUT_CODE	Y	11200	

\*Indicates a required column

## SUT Apply Codes

The SUT\_APPLY table contains an entry for all the Sales Use Tax Apply Codes.

	A	B	C	
1	<b>Table name*</b>	<b>Active*</b>	<b>Apply Code*</b>	
2	SUT_APPLY	Y	YES	
3	SUT_APPLY	Y	NO	

\*Indicates a required column

## VAT Code

The VAT\_CODE table provides a mapping between the Jurisdiction, the VAT Code and the VAT Rate. Each rate requires a combination of Jurisdiction and VAT Code. Jurisdiction is not optional, but you may use a default value if the Jurisdiction never varies. For example, if your solution only processes local invoices, you could provide a single default value throughout the Jurisdiction column. Provide VAT rates in the table as a percentage between 0 and 100. For example, enter 7.5% as 7.5, rather than .75.

For an overview of incorporating VAT into the eForm, refer to the [Configure value added tax \(VAT\) within the eForm](#) section of this document.

	A	B	C	D	E	
1	<b>TableName*</b>	<b>Jurisdiction*</b>	<b>VATCode*</b>	<b>VATRate*</b>	<b>Description</b>	
2	VAT_CODE	FR	V0	0	France zero tax rate	
3	VAT_CODE	FR	V1	19.6	France full tax rate	
4	VAT_CODE	FR	V2	7	France reduced tax rate	
5	VAT_CODE	FR	V7	0	France tax exempt, foreign vendor	

\*Indicates a required column

## Vendor Location

The VENDOR\_LOCATION table contains a location value that relates to the vendor. PeopleSoft implementations primarily use this table.

	A	B	C	D	E	
1	<b>TableName*</b>	<b>VendorID*</b>	<b>VendorLoc_Code*</b>	<b>VendorLoc_Name*</b>	<b>Active*</b>	
2	VENDOR_LOCATION	1	P	PRIMARY	A	
3	VENDOR_LOCATION	2	P	PRIMARY	A	
4	VENDOR_LOCATION	3	P	PRIMARY	A	
5	VENDOR_LOCATION	3	S	SECONDARY	A	

\*Indicates a required column

## VAT Registration Number (VAT ID) Vendor

The VAT\_ID\_VENDOR table provides the ability to relate one or more VAT registration numbers (VAT IDs) per vendor. For example, in the following virtual table sample, there are two VAT IDs associated with VendorID 1000.

	A	B	C	D	E	
1	<b>TableName*</b>	<b>Active*</b>	<b>BusUnitId*</b>	<b>VendorID*</b>	<b>VAT_ID*</b>	
2	VAT_ID_VENDOR	Y	1234	1000	ATU12345678	
3	VAT_ID_VENDOR	Y	1234	1000	DK12345678	
4	VAT_ID_VENDOR	Y	1234	1001	HU12345678	
5	VAT_ID_VENDOR	Y	1234	1002	IT12345678901	
6	VAT_ID_VENDOR	Y	2468	6000	PT123456789	
7	VAT_ID_VENDOR	Y	2468	6000	ESX1234567X	

\*Indicates a required column

## VAT Registration Number (VAT ID) Business Unit

The VAT\_ID\_BUS\_UNIT table provides the ability to relate one or more VAT registration numbers per business unit. For example, in the following virtual table sample, there are four VAT IDs associated with BusUnitID 1234.

	A	B	C	E
1	Table Name*	Active*	Bus Unit ID*	VAT ID*
2	VAT_ID_BUS_UNIT	Y	1234	ATU12345678
3	VAT_ID_BUS_UNIT	Y	1234	DK12345678
4	VAT_ID_BUS_UNIT	Y	1234	HU12345678
5	VAT_ID_BUS_UNIT	Y	1234	IT12345678901
6	VAT_ID_BUS_UNIT	Y	2468	PT123456789
7	VAT_ID_BUS_UNIT	Y	2468	ESX1234567X

\*Indicates a required column

## Virtual Table Updater configuration file

The columns in the CSV table files match the <column> entries within each of the record structures in the AP\_VirtualTable\_Updater.xml file. This configuration file determines how you update virtual tables with business application data. Use the AP\_VirtualTable\_Updater.js script on a regular basis to update the information between the virtual tables in the ImageNow database and the business application.

The AP\_VirtualTable\_Updater.xml file includes a record structure for each of the CSV table files created during installation. You can switch the order of the record structures and of the columns within each record structure. However, the column order must be consistent between the associated CSV file and the AP\_VirtualTable\_Updater.xml file. For any optional column that your specific solution does not require, in the column element, set enabled to "false."

If you work with dates, specify the date format in one of the following formats:

- MMDDYY
- MMDDYYYY
- DDMMYY
- DDMMYYYY
- YYMMDD
- YYYYMMDD

If not specified, the default format is MMDDYYYY.

Example:

```
<column customprop="Z_APW PO Date" enabled="true" format="YYYYMMDD" />
```

### AP\_VirtualTable\_Updater.xml

The example below demonstrates the record structure for the PO virtual table.

```
<recordstructure recordtype="PO" table="Z_APW_APPO" enabled="true">
  <column customprop="Z_APW Business Unit ID" enabled="true"/>
  <column customprop="Z_APW PO Number" enabled="true"/>
  <column customprop="Z_APW Vendor ID" enabled="true"/>
  <column customprop="Z_APW PO Amount" enabled="true"/>
  <column customprop="Z_APW PO Currency" enabled="true"/>
  <column customprop="Z_APW PO Date" format="YYYYMMDD" enabled="true"/>
  <column customprop="Z_APW PO Date Closed" enabled="true"/>
  <column customprop="Z_APW Vendor Remit To" enabled="true"/>
</recordstructure>
```

When you run the Virtual Table Updater, it detects the CSV files, processes and loads the data into the virtual tables, and moves the CSV files to the [drive:]inserver6\temp\ap\archive directory.

Before you run the Virtual Table Updater, you must verify the settings in the updater XML configuration file and place the CSV files in the correct directories. In the configuration file, the input path, archive path, and error path are hardcoded to the C:\ drive.

**Important** If ImageNow Server is not on the C:\ drive, you use a non-Windows platform, or have an active-active server environment, you must change this setting in the XML configuration file.

## Virtual tables updates

You need to determine the frequency and method for using the updater script file to update your records between the virtual tables in ImageNow and your business application. Understanding how this process works can help you decide how to employ the updating process for your implementation. From INTTool, you run the AP\_VirtualTable\_Updater.js file to complete an incremental update. The AP\_VirtualTable\_Updater.js file follows the configuration options set in the AP\_VirtualTable\_Updater.xml file when processing the request.

When you run the script, it inspects each record in the virtual tables. If a record in the business application (CSV file) matches a record that exists in the virtual table, the script updates the record in the virtual table from the business application record. The script can update any of the columns in the virtual table, including the Active flag. If the matching record is inactive, the script marks the virtual table record as inactive and considers it a soft delete. If a record that exists in the business application does not match a record in the virtual table, the script inserts (adds) a new record in the virtual table. It is highly recommended to add a task to the Windows Scheduler to run the script. For example, you could set the script to run on an hourly, daily, or weekly basis. We also recommend that you schedule the script to run after hours, when overall system utilization is low. You can also use other methods of automating the script, but consider it a best practice to use Windows Scheduler. Your vendor list may only need to be refreshed nightly, whereas changes to the PO table may require immediate updates.

## Archive processed files

You can configure the AP\_VirtualTable\_Updater.js script to archive processed files automatically to a specified directory. Beginning with version 10.4.1.x, the AP\_Config.js script can append the processed filename with a time stamp. For example, *<file name>\_YYYYMMDD\_hhmmss.csv*. The Virtual Table Updater cannot move a file with a duplicate name to the archive directory.

To enable the time stamp, complete the following step.

1. Navigate to **[drive:]\\server6\\etc\\ap** and open **AP\_Config.xml** with a text editor.
2. Under the **<AP\_VirtualTable\_Updater>** node, set **<AppendDateTimeStampToFile>** to **true**.

## Document types and custom properties

The installation process creates two document types, Invoice and Credit Memo, and all the custom properties used in conjunction with the AP Invoice eForm solution. The following tables describe the custom properties associated with the document types.

### Invoice

Custom Property Name	Data Type	Description
Company	String	The business unit for the invoice
Vendor ID	String	The unique identification number for the vendor associated with the invoice
Vendor Name	String	The name of the vendor associated with the invoice
Invoice Number	String	The unique number associated with the invoice
Invoice Date	Date	The invoice issue date
Invoice Due Date	Date	The invoice due date
Invoice Amount	Number	The invoice total amount
Invoice Tax Amount	Number	The invoice total tax amount
Pay Immediately	Flag	A flag that indicates whether the invoice is processed immediately
PO Number	String	The unique purchase order number associated with the invoice
PO Amount	Number	The purchase order amount associated with the invoice
PO Variance Amount	Number	The amount, in dollars, for variance exceptions
PO Variance Factor	Number	The amount, in percentages, for variance exceptions
AP Header Valid	String	The header error codes used for exception handling
AP Detail Valid	String	The detail error codes used for exception handling
AP Custom Valid	String	The custom error codes used for exception handling
Invoice Export Status	String	The invoice export status

## Credit Memo

Custom Property Name	Data Type	Description
Company	String	The invoice business unit
Vendor ID	String	The unique identification number for the vendor associated with the invoice
Vendor Name	String	The name of the vendor associated with the invoice
Credit Memo Number	String	The unique number associated with the credit memo
Credit Date	Date	The credit memo creation date
Credit Amount	Number	The total amount of the credit memo
Credit Tax Amount	Number	The total amount of taxes of the credit memo
Pay Immediately	Flag	Flag that indicates whether the invoice is processed immediately
PO Number	String	The unique purchase order number associated with the invoice
PO Amount	Number	The purchase order amount associated with the invoice
PO Variance Amount	Number	The amount, in dollars, for variance exceptions
PO Variance Factor	Number	The amount, in percentages, for variance exceptions
AP Header Valid	String	The header error codes used for exception handling
AP Detail Valid	String	The detail error codes used for exception handling
AP Custom Valid	String	The custom error codes used for exception handling
Invoice Export Status	String	The invoice export status
Invoice Reference Number	String	The invoice number that the credit memo refers to
Credit Reason	String	The reason for the credit memo issuance

## AP Invoice eForm file directories

The following table details the directories of all of the AP Invoice eForm files that the system installs when you run the AP Invoice eForm installation wizard.

### \inserver6\etc\ap

Installed Form File	Description
AP_Config.xml	This XML file stores the business application name. Leave the <virtualtable> attribute value as true unless this is a custom form implementation. Valid entries include Lawson, PeopleSoft, and Great Plains. The selected business application appears in this XML file in the <host> element. You can use the eForm for virtually any business application that receives uploaded invoices. To install the eForm for any other business application, choose one of the existing available business application options that best fits the scenario.
AP_Custom_Cultures.xml	This XML file contains the configuration information used for global customization in your AP Invoice eForm.
AP_Export.xml	This XML file contains the configuration information used when you are exporting data from ImageNow to your business application.
AP_Skin.xml	This file is the main configuration file for the invoice form. For complete configuration details, refer to the <a href="#">AP_Skin.xml</a> section in this document for complete configuration details.
AP_Validate_Server.xml	This XML file contains the data-mapping configuration between your form and the custom properties and document keys in ImageNow. Use this file to configure which fields are required user entries in the form.
AP_VirtualTable_Updater.xml	This XML file contains the configuration information used for importing data from your business application, via CSV data files, into the virtual tables.

## \inserver6\temp\ap

If your ImageNow Server is not on the C:\ drive, you use a non-Windows platform, or you have an active-active server environment, then you must configure the directory paths in the **AP\_VirtualTable\_Updater.xml** and **AP\_Export.xml** files. The tables below outline the directories as they appear by default in **AP\_VirtualTable\_Updater.xml**.

Default paths:

Default Directory	Description	File	Attribute
C:\inserver6\temp\ap\archive	The location where the AP_VirtualTable_Updater.js script sends the CSV data files after processing.	AP_VirtualTable_Updater.xml	archivepath
C:\inserver6\temp\ap\error	The location where the AP_VirtualTable_Updater.js script sends the CSV files if it encounters errors.	AP_VirtualTable_Updater.xml	errorpath
C:\inserver6\temp\ap\export	The location where the AP_VirtualTable_Updater.js script sends the exported data files for upload into your business application.	AP_Export.xml	outputdir
C:\inserver6\temp\ap\import	The location from which the AP_VirtualTable_Updater.js script imports CSV files.	AP_VirtualTable_Updater.xml	inputpath

## \inserver6\script

Installed Form File	Description
AP_Custom_Invoice_Export.js	The form uses this script to massage export data. This script should contain all custom functions you need to export valid data.
AP_Custom_Validate_Server.js	The form uses this script when you write custom validation logic.
AP_DupCheck.js	The form uses this script to execute duplicate invoice logic. The configuration section for duplicate check is located in the AP_Config.xml file.
AP_GetGLAcctCode.js	The form uses this script to retrieve GL account codes.
AP_GetCompanies.js	<p>The form uses this script to retrieve Business Unit, SUT, Currency, Location, Special Handling codes, and other information.</p> <p>To improve performance while loading the eForm, the eForm caches the drop-down values that the AP_GetCompanies.js script gathers from the virtual tables. The script saves the cached file in [drive:]\inserver6\temp\AP_GetCompanies.cache.</p> <p>The script runs every time the eForm loads. Each time, the script settings determine whether to use cached values or load values from the virtual tables. The following configurable parameter determines when the cached values become stale.</p> <pre>#define CACHE_DURATION_MINUTES 60</pre> <p>By default, the script caches these values for 60 minutes. You can adjust this time as necessary.</p> <p><b>Important</b> If you are testing AP_VirtualTable_Updater.js and do not see the changes (new values) reflected in the AP Invoice eForm, delete the cached file from [drive:]\inserver6\temp. Removing this temporary file refreshes the cached data.</p>
AP_GetData.js	The form uses this script to retrieve PO information related to the Credit invoice type.
AP_GetData_Custom.js	The form uses this script to get your custom data, if any.
AP_GetGLAcctCodes.js	The form uses this script to retrieve GL Codes.
AP_GetGLCodeLists.js	The form uses this script to retrieve a GL Code list from the GL Codes table.
AP_GetGLCodeUserConstraints.js	The form uses this script to retrieve the GL User constraints per GL Codes.
AP_GetPO.js	The form uses this script to retrieve Purchase Order information given a vendor ID or vendor name.
AP_GetPOLines.js	The form uses this script to retrieve Purchase Order information.
AP_GetVendorLoc.js	The form uses this script to retrieve vendor location information.

Installed Form File	Description
AP_GetVendors.js	The form uses this script to retrieve Vendor Information for the vendor list in the Vendor Search dialog box.
AP_Invoice_Export.js	ImageNow uses this script to export XML data from ImageNow into the appropriate format for the related business application.
AP_Maintenance.js	The form uses this INTool script to complete client side caching and other maintenance tasks associated with performance. We recommend that you schedule this script to every minute.
AP_ReindexAndCPMapping.js	Use this script in ImageNow workflow to index form data and set form data into custom properties based on the AP_Validate_server.xml file configuration.
AP_ValidateGLAcctCodes.js	Use this script to validate GL Account code entries. The script is executed when the user clicks the VALIDATE button on the GL Entry portion of the form.
AP_Validate_Server.js	Use this script in ImageNow workflow to validate form values, retrieve data, and populate custom properties or document key values. Apply the script as an in-bound action.
AP_ValidateGLCodes.js	The form uses this script to validate GL code entries. The script runs when the user clicks the VALIDATE button on the GL Entry portion of the form.
AP_VirtualTableInstaller.js	The installer uses this script to install the virtual tables.
AP_VirtualTable_Purge.js	INTool calls this whenever the Purge routine needs to be run against the virtual table data.
AP_VirtualTable_Updater.js	Use this script to update your virtual tables with the business application data from the CSV files.

## \inserver6\form\[build folder]

The build folder name appears in a format similar to 2000000003\_00006T3XEMH3. You can locate the following files in the \inserver6\form\[build folder] directory.

**Note** You may also see the build folder path as \inserver6\worksheet\[build folder].

Installed Form File	Description
AP.xsl	This file is the presentation file for the AP Invoice eForm. Do not modify this file. Instead, use the AP_Skin.xml file and the WorksheetStylesheet.css file to alter the form format and presentation.
AP_CustomValidation_Client.js	This file contains any custom logic that you might add as form event handlers. For example, when the user tabs off of an eForm field, you might want to add some additional processing logic. The logic, implemented as a JavaScript function, would be placed in this file.
AP_Worksheet_Scripts.js	ImageNow uses this script within the form to provide search dialogs for looking up general ledger account codes and looking up vendors.  The form uses this script to apply the configurations contained within the AP_Skin.xml file to the form display and to validate form data.
AP_SkinCache.js	This local cache file loads inline logic to the invoice form when the form opens.
AP_Skin.xml	This file contains a cached copy of the configuration settings AP.xsl uses.
<customerlogo>.jpg	This is the customer logo image file used in the invoice form. This file name must match the customerlogo entry in the AP_Skin.xml file.
globalize.js	This file contains all the generic globalization functions used by the AP Invoice eForm. This file also contains the default values for the en-US language setting for when the system does not detect a valid language. Do not modify this file.
globalize_culture_all.js	This file contains the culture information for all supported cultures. Do not modify this file.
globalize_culture_custom.js	This file contains customization configurations to cultures or newly defined cultures. Do not modify this file.
WorksheetStylesheet.css	This is the cascading style sheet file used for the presentation of the AP Invoice eForm. Use this file to modify the styles used in your form implementation.

## \inserver6\form\data\_definition

The following XML files are located in the \inserver6\form\data\_definition directory.

**Note** You may also see this folder path as \inserver6\worksheet\data\_definition.

Installed Form File	Description
AP_Invoice.xml	This file is the data definition file for the AP Invoice eForm and typically should not need modification. The only modification that is allowed is the default setting for the invoice type (PO or non-PO).

## AP Invoice eForm log files

Log files associated with the invoice form installation and configuration are stored in the \inserver6\log directory.

Installed Form File	Description
AP_GetCompanies_<current date>.log	This log file is created after you open the form.
AP_GetVendors_<current date>.log	This log file is created after you run a vendor search in the form.
AP_VirtualTable_Installer_<current date>.log	This log file is created after you run the installer. The file details the success or failure of the virtual table creation during installation.
AP_VirtualTable_Updater_<current date>.log	This log file is created after you run the AP_VirtualTable_Updater.js script. The log file details the success or failure of the virtual table updates between ImageNow and the CSV data files the customer places in the path specified in the input path variable in the ap_virtualtable_updater.xml file.

## AP Invoice eForm configuration files

The following tables detail each of the XML configuration files included with your AP Invoice eForm installation.

### AP\_Skin.xml

This file is located in the [drive:]\inserver6\etc\ap directory.

Element	Description
<configuration>	Root element.
<defaultlanguage></defaultlanguage>	<p>This global element allows you to specify the country-locale in which you want to display the AP Invoice eForm.</p> <p>The default is blank.</p>
<customerlogo> <i>logo object name, for example, psi.jpg</i> </customerlogo>	<p>A value for this element is optional. Replace this file name with your logo file name or URL. If the value has a URL and file name, there is white space in the header of the form. If the value has a path and graphic, the graphic displays in the middle of the header of the AP Invoice eForm.</p> <p>To use a graphic file, you must add the graphic file to the AP Invoice eForm presentation in ImageNow. This image file must exist in the build folder within the [drive:]\inserver6\worksheets directory. Alternatively, you can use a value for a logo or graphic located on the Internet or an intranet.</p>
<changetabs>true</changetabs>	<p>This setting allows you to lock down the PO or non-PO value in the list box if you process only one type of invoice in association with the form. This attribute uses the default value set in the AP_Invoice.xml file. If the value is set to true, you can change the value. If the value is set to false, the list box is locked.</p>
<validatebutton>true</validatebutton>	<p>This required element is set to true by default. When this value is set to true, you make the Validate button available to users who have any of the following privileges:</p> <ul style="list-style-type: none"> <li>• read/write access to invoice header information</li> <li>• read/write access to GL coding lines (non-PO invoices)</li> <li>• read/write access to PO lines (PO-based invoices)</li> </ul>
<duplicateinvoicecheck>true</duplicateinvoicecheck>	<p>This setting allows you to specify whether to enable client-side duplicate invoice checking.</p> <p>The default is true.</p>
<vatsummary>	<p>If your AP Invoice eForm solution uses value added tax (VAT), configure this section.</p>

Element	Description
<enable>>false</enable>	This setting enables VAT Summary support in the AP Invoice eForm. The default is false.
<showsection>>false</showsection>	This setting displays the VAT Summary section in the AP Invoice eForm. The default is false.
<validatesummary>>false</validatesummary>	This setting enables validation of the line-level VAT Amount fields' sum against the header-level VAT Total field. The default is false.
</vatsummary>	
<po>	
<polines>>true</polines>	This element requires a value. If configuring a PO-related eForm, to show and modify PO lines on the eForm set to true.
<gllines>>false</gllines>	This element requires a value. This function exists because backwards-compatibility is not available. This attribute should be left as false.
<change_remitto>>true</change_remitto>	To permit the user to change the Vendor Remit To field for PO-based invoices, set to true.
<distributiontotal>	Each element within the <distributiontotal> element requires a value. This element specifies which charges to include in the To Be Distributed calculation.
<include_freight>>true</include_freight>	This element specifies whether to include freight charges in the To Be Distributed calculation. For PO-based invoices, the default value is true.
<include_tax>>true</include_tax>	This element specifies whether to include tax charges in the To Be Distributed calculation. For PO-based invoices, the default value is true.
<include_misc>>true</include_misc>	This element specifies whether to include miscellaneous charges in the To Be Distributed calculation. For PO-based invoices, the default value is true.

Element	Description
<code>&lt;include_addl_amt1&gt;&gt;false&lt;/include_addl_amt1&gt;</code>	<p>This element specifies whether to include additional charges in the To Be Distributed calculation.</p> <p>The default value is false.</p> <p>Note You can use the following format to configure up to eight additional charges fields:</p> <pre> &lt;include_addl_amt1 &gt;&gt;false&lt;/include_addl_amt1&gt; &lt;include_addl_amt2 &gt;&gt;false&lt;/include_addl_amt2&gt; . . . &lt;include_addl_amt8 &gt;&gt;false&lt;/include_addl_amt8&gt; </pre>
<code>&lt;/distributiontotal&gt;</code>	
<code>&lt;/po&gt;</code>	
<code>&lt;nonpo&gt;</code>	
<code>&lt;distributiontotal&gt;</code>	<p>AP_Skin.xml requires a value for each element within the <code>&lt;distributiontotal&gt;</code> element. This element specifies which charges to include in the To Be Distributed calculation.</p>
<code>&lt;include_freight&gt;&gt;false&lt;/include_freight&gt;</code>	<p>This element specifies whether to include freight charges in the To Be Distributed calculation.</p> <p>The default value is false.</p>
<code>&lt;include_tax&gt;&gt;false&lt;/include_tax&gt;</code>	<p>This element specifies whether to include tax charges in the To Be Distributed calculation.</p> <p>For non-PO invoices, the default value is false.</p>
<code>&lt;include_misc&gt;&gt;false&lt;/include_misc&gt;</code>	<p>This element specifies whether to include miscellaneous charges in the To Be Distributed calculation.</p> <p>For non-PO invoices, the default value is false.</p>

Element	Description
<include_addl_amt1>false</include_addl_amt1>	<p>This element specifies whether to include additional charges in the To Be Distributed calculation.</p> <p>The default value is false.</p> <p>Note You can use the following format to configure up to eight additional charges fields:</p> <pre>&lt;include_addl_amt1 &gt;false&lt;/include_addl_amt1&gt; &lt;include_addl_amt2 &gt;false&lt;/include_addl_amt2&gt; . . . &lt;include_addl_amt8 &gt;false&lt;/include_addl_amt8&gt;</pre>
</distributiontotal>	
</nonpo>	
<credit>	
<distributiontotal>	<p>AP_Skin.xml requires each element within the &lt;distributiontotal&gt; element. This element specifies which charges to include in the To Be Distributed calculation.</p>
<include_freight>true</include_freight>	<p>This element specifies whether to include freight charges in the To Be Distributed calculation.</p> <p>The default value is false.</p>
<include_tax>true</include_tax>	<p>This element specifies whether to include tax charges in the To Be Distributed calculation.</p> <p>For credit-based invoices, the default value is false.</p>
<include_misc>true</include_misc>	<p>This element specifies whether to include miscellaneous charges in the To Be Distributed calculation.</p> <p>For credit-based invoices, the default value is false.</p>

Element	Description
<code>&lt;include_addl_amt1&gt;&gt;false</code> <code>&lt;/include_addl_amt1&gt;</code>	<p>This element specifies whether to include additional charges in the To Be Distributed calculation.</p> <p>The default value is false.</p> <p>Note You can use the following format to configure up to eight additional charges fields:</p> <pre> &lt;include_addl_amt1 &gt;&gt;false&lt;/include_addl_amt1&gt; &lt;include_addl_amt2 &gt;&gt;false&lt;/include_addl_amt2&gt; . . . &lt;include_addl_amt8 &gt;&gt;false&lt;/include_addl_amt8&gt; </pre>
<code>&lt;/distributiontotal&gt;</code>	
<code>&lt;/credit&gt;</code>	
<code>&lt;polines&gt;</code>	
<code>&lt;addbutton&gt;&gt;false&lt;/addbutton&gt;</code>	<p>If true, this required element allows users to add PO lines.</p> <p>The default value is false.</p>
<code>&lt;delbutton&gt;&gt;false&lt;/delbutton&gt;</code>	<p>If true, this required element allows users to delete PO lines.</p> <p>The default value is false.</p>
<code>&lt;/polines&gt;</code>	
<code>&lt;gllines&gt;</code>	
<code>&lt;addbutton&gt;&gt;true&lt;/addbutton&gt;</code>	<p>If true, this required element allows users to add GL lines.</p> <p>The default value is false.</p>
<code>&lt;linestoadd&gt;1&lt;/linestoadd&gt;</code>	<p>This element specifies the number of lines added to the eForm when the user clicks the Add Line button on the user interface.</p>
<code>&lt;delbutton&gt;&gt;true&lt;/delbutton&gt;</code>	<p>If true, this required element gives users the ability to delete GL lines.</p> <p>The default value is true.</p>
<code>&lt;linewrapping&gt;true&lt;/linewrapping&gt;</code>	<p>This element specifies whether GL code fields wrap to additional lines or remain displayed on a single line.</p> <p>The default value is true.</p> <p><b>Note</b> If you enable line wrapping, you can define which line each field appears under with the <code>&lt;glline label/&gt;</code> element.</p>

Element	Description
</gllines>	
<readonlyifexported>true</readonlyifexported>	<p>If true, this required element prevents users from exporting data more than one time, either by accident or intent. It also prevents the user from modifying a previously exported invoice. The user receives an error message when attempting to export a second time.</p> <p>The default value is true.</p>
<errorpopup>true</errorpopup>	<p>If true, this required element allows users to receive an error message containing a field validation error, such as an invalid date or amount.</p> <p>If false, the cursor stays on the input field until the user enters a valid value, and the user does not receive an error message.</p> <p>The default value is true.</p>
<roles>	<p>To configure the sections of the AP Invoice eForm that users and groups can view and modify, use these roles.</p> <p>You configure four user roles in the &lt;roles&gt; element. Specify one of the following values for the detail attribute of each user role, as outlined in the <a href="#">About workflow queue roles</a> section.</p> <p>hidden: Use this option if the data does not display to the user in the AP Invoice eForm for that role.</p> <p>readonly: Use this option if the data displays but the user cannot modify it in the form.</p> <p>enabled: Use this option if the data displays the user can modify it in the</p>
<role name="Invoice Entry" header="enabled" detail="hidden" po_detail="hidden" gl_detail_po="hidden" gl_detail_nonpo="hidden">	<p>The Invoice Entry role is for the user or group who enters invoice header data. In most cases, the header is set to enabled and the detail is hidden. This is because only the header data is being entered during this role of the process.</p> <p>You can configure the po_detail, gl_detail_po, and gl_detail_nonpo attributes in addition to, or instead of, the detail attribute. Note that the po_detail, gl_detail_po, and gl_detail_nonpo configuration values take precedence over the detail value.</p>
<queue invoicetype="nonpo" changetabs="true">APO Invoice Entry</queue>	<p>The queue value in this entry and in ImageNow Workflow must match exactly. Add an entry for each workflow queue that completes the ENTRY function of the process.</p> <p>The invoicetype attribute specifies the default invoice type for the queue (nonpo or po). The changetabs attribute specifies whether a user of the queue can modify the invoice type.</p>

Element	Description
</role>	
<role name="Invoice Processing" header="enabled" detail="enabled" po_detail="enabled " gl_detail_po="enabled " gl_detail_nonpo="enabled ">	The Invoice Processing role handles GL coding entry for Non-PO Invoices. This role also handles Match Audit Exceptions for PO-based invoices. In most cases, the header is enabled and the detail is enabled. This is because, while the primary purpose of this role is to enter detail data, there is still a chance that header data might need modification.
<queue invoicetype="nonpo" changetabs="true">APO Invoice Processing</queue>	The queue value in this entry and in ImageNow must match exactly. Add an entry for each workflow queue that completes the PROCESSING function of the process.  The invoicetype attribute specifies the default invoice type for the queue (nonpo or po). The changetabs attribute specifies whether a user of the queue can modify the invoice type.
</role>	
<role name="Invoice Approval" header="readonly" detail="enabled" po_detail="enabled " gl_detail_po="enabled " gl_detail_nonpo="enabled ">	The Invoice Approval role is for users and groups who complete invoice approval on non-PO invoices. In most cases, the header is readonly and the detail is enabled. This is because, during the approval process, the header information should not be changed. However, the approver might need to modify the detail data. In some cases, you might elect to prevent the approver from modifying anything. In this case, set the detail to be readonly.
<queue invoicetype="nonpo" changetabs="true">APO Invoice Approval</queue>	The queue value in this entry and in ImageNow Workflow must be exactly the same name. Add an entry for each workflow queue that completes the APPROVAL function of the process.  The invoicetype attribute specifies the default invoice type for the queue (nonpo or po). The changetabs attribute specifies whether a user of the queue can modify the invoice type.
</role>	
<role name="(default)" header="readonly" detail="readonly" po_detail="enabled " gl_detail_po="enabled " gl_detail_nonpo="enabled ">	The default user role is for users and groups who need access from workflow queues other than the ones defined in the other roles. You can also use this role for users or groups who need access to the form outside of workflow, such as in the Documents pane.
</role>	
</roles>	

Element	Description
<fields>	<p>The &lt;fields&gt; element configures the form fields and labels. We advise against changing these elements without training. Consult a scripting engineer at Perceptive Software to determine the impact of any changes. Do not add to, or delete, child elements in the &lt;fields&gt; element; instead, use the disable attribute. You should thoroughly test any changes you make to this element.</p>
<header/>	<p>The header label information includes mapped values that should not be changed. You can configure the visible, width, disabled, maxlength, align, title, distributable, required, and aoc attributes. Do not change the label or field attributes; instead, change the caption attribute.</p> <p><b>Notes</b></p> <p>&lt;distributable&gt; is only applicable for the fields, Sales Tax, Freight Amount, Misc Amount, and the eight Additional Amount fields.</p> <p>If &lt;distributable&gt; is set to true, the system displays an ellipsis button beside the input field. Click this button to set focus to the first editable field in the appropriate related GL Distribution section.</p> <p>To associate a default AOC Code with a Sales Tax, Freight, Misc, or Additional Amount&lt; 1-8&gt; field, provide the AOC Code from your <a href="#">AOC GL Code</a> or <a href="#">AOC GL Account</a> virtual tables. For example, if you set &lt;header label="lblAddlAmt1" aoc="BI"&gt;, when the user types a value in the Addl Amt 1 field, the default information associated with AOC Code "BI" populates the line-level GL Distributions section of the eForm.</p>
<poline/>	<p>The PO line label information includes mapped values that should not be changed. You can change the visible, width, disabled, maxlength, align, title, and the required attributes. Do not change the label or field attributes; instead, change the caption attribute.</p>
<glline/>	<p>The general ledger line label information includes mapped values that you should not change. You can change the freeform, visible, width, disabled, maxlength, align, title, and the required attributes. Do not change the label or field attributes; instead, change the caption attribute.</p> <p>If the freeform attribute is set to true, the form will not use the GL Code lookup feature. Set the freeform attribute to false to use the GL Code lookup feature.</p> <p>If &lt;linewrapping&gt; is set to true, you can specify the row on which the GL field appears by changing the line attribute. For example, line="2" indicates the field appears on the second line.</p>

Element	Description
<addlentries/>	<p>The labels for the Additional Entries sections. This includes Additional Entry Description, Additional Entry Amount, and AOC Code.</p> <p>To enable the AOC list for use with PO GL Distributable fields, for the element labeled lblAOCCode, set the visible attribute to true.</p>
<vatsummary/>	The labels for the VAT Summary section.
<footer/>	The footer label information includes mapped values that should not be changed. You can change the visible, width, disabled, maxlength, align, title, and the required attributes. Do not change the label or field attributes; instead, change the caption attribute.
</fields>	
<conditionalfields>	The <conditionalfields> element configures the form fields and labels and overrides the attribute settings defined in the <fields> section.
<header label="lblHdr< 1-10>" field="txtHdr< 1-10>" visible="true captionkey="LBL_HDR_< 1-10>:" invoicetype="nonpo"/>	The header information includes mapped values visible, caption, and invoicetype. Do not change the header label: instead, change the captionkey attribute or the underlying caption for the captionkey attribute. The invoice type attribute specifies which type of invoice to apply to these conditional attributes (non-PO, PO, or credit memo).
</conditionalfields>	
</configuration>	

## AP\_Config.xml

This file is stored in the \inserver6\etc\ap directory.

Element	Description
<configuration>	Root element.
<host>peoplesoft</host>	This required element is set based on your selection in the installation wizard. Valid entries include lawson, peoplesoft, and greatplains. The selected business application is listed in this XML file. You can use the eForm for virtually any business application that receives uploaded invoices. To install the eForm for any other business application, choose the business application option that best fits the scenario.
<datemask>MMDDYYYY</datemask>	This element specifies the default date format for the Invoice Date entry. Valid formats are MMDDYYYY, DDMMYYYY, and YYYYMMDD. Slashes are optional. If not specified, the default format is MMDDYYYY.
<dupcheck enabled="true">	This element specifies whether to check for duplicates in iScript (AP_ValidationServer.js).  The default is true.

Element	Description
<code>&lt;dupcheck_vsl enabled="true"&gt;[tab] =%s AND [f3] = %s&lt;/dupcheck_vsl&gt;</code>	<p>This element contains the VSL query statement for detecting duplicate invoices. The first %s is replaced with Vendor ID, and the second %s is replaced with Invoice Number. The default VSL statement assumes the document key Field2 or tab contains the Vendor ID and the document key Field3 contains the Invoice Number. Any changes to these assumptions require modification to the VSL statement.</p> <p>Duplicate invoices are detected in two places in the application:</p> <p>When the user tabs off the Invoice Number, a call is made to an iScript that searches ImageNow documents for potential matches based on a combination of Vendor ID and Invoice Number. You can modify the VSL query if necessary. For example, you can modify the query to specify the Drawer or Document Type in addition to Vendor ID and Invoice Number.</p> <p>The inbound iScript (AP_Validate_Server.js) completes the same check for duplicates between Vendor ID and Invoice Number using the same VSL query. If the script finds any duplicate documents, it sets the custom property "AP Header Valid" to "ERR_DUP_INVOICE".</p> <p>The script does not route the duplicate document back to the previous queue. Instead, the routing rules route these types of documents based on the value of this custom property.</p>

Element	Description
<code>&lt;statement view="All Documents" match="2" enabled="false"&gt;VendorID   InvoiceNumber   TotalAmount # 5 &lt;/statement&gt;</code>	<p>This element contains advanced logic for duplicate invoice check.</p> <p>The statement view attribute specifies the custom view that you can configure to match against literals.</p> <p>The statement match attribute specifies how many of the values within the statement the query must match.</p> <p>The % specifies within what percentage range the amount must fall for the query to match.</p> <p>The # specifies within what amount range the amount must fall for the query to match.</p> <p>Valid operators:</p> <p>% (+/- percent value)</p> <p># (+/- amount value)</p> <p><b>Note</b> You can use a custom view for comparisons of literals.</p> <p>Valid keys:</p> <p>CompanyNumHdr, LocationCode, PONumber, VendorID, InvoiceNumber, InvoiceDate, TotalAmount, InvoiceReferenceNumber, TaxAmount, FreightAmount, MiscCharge, AddlAmt&lt;1 to 8&gt;, Hdr&lt;1 to 8&gt;</p>
<code>&lt;/dupcheck&gt;</code>	

Element	Description
<polinematching>true</polinematching>	<p>This element specifies whether to complete PO matching at the line-item level or the header level.</p> <p>If <b>true</b>, this element indicates that PO matching occurs at the line-item level. The BW_ReindexDoc.js script matches Intelligent Capture-extracted PO Line numbers with values in the PO Line virtual table.</p> <p>If the script finds a match, then it populates the eForm with the following values from the virtual table.</p> <ul style="list-style-type: none"> <li>• PO Line Number</li> <li>• PO Quantity</li> <li>• PO Unit Price</li> <li>• Unit of Measure</li> <li>• VAT Jurisdiction</li> <li>• VAT Code</li> <li>• VAT Amount</li> </ul> <p>In the eForm, the user can compare the scanned values against the PO values.</p> <p>If <b>false</b>, this element indicates that PO matching occurs at the header level. The AP_Validate_Server.js script matches the Intelligent Capture-extracted PO number with the value in the PO Line virtual table. The script then loads all open PO lines into the eForm.</p> <p><b>Note</b> In the eForm, users can also manually load PO lines from the virtual table.</p> <p>If &lt;polinematching&gt; is true, in the AP_Skin.xml file, set the visible attribute for each of the following elements to true. If &lt;polinematching&gt; is false, in AP_Skin.xml, set the visible attributes to false.</p> <ul style="list-style-type: none"> <li>• txtPO_LineNumber</li> <li>• txtPO_Quantity</li> <li>• txtPO_UOM</li> <li>• txtPO_UnitPrice</li> <li>• txtPO_VATJurisdiction</li> <li>• cboPO_VATCode</li> </ul>
<viewname>Z_APW All Projects</viewname>	<p>The viewname element specifies the name of the custom view you created during the Accounts Payable Invoice eForm installation and setup process.</p>
<scripts>	<p>This element contains the scripts used by the form. For more information on each script, refer to the <a href="#">\inserver6\script</a> section in this document.</p>

Element	Description
<AP_DupCheck>	
<LogToFile type="boolean">false</LogToFile>	<p>This element sets the logging for AP_DupCheck.js to the console or to a log file.</p> <p>false = the console</p> <p>true = a log file</p> <p>The default is false.</p>
<DebugLevel type="int">4</DebugLevel>	<p>The logging level for AP_DupCheck.js.</p> <p>Use minimal logging after setup, unless you are debugging an issue. If you increase the logging level, reset it after you finish debugging. Failure to do so can slow performance and consume hard disk space.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
</AP_DupCheck>	
<AP_GetCompanies>	
<LogToFile type="boolean">false</LogToFile>	<p>This element sets the logging for AP_GetComapnies.js to the console or to a log file.</p> <p>false = the console</p> <p>true = a log file</p> <p>The default is false.</p>
<DebugLevel type="int">4</DebugLevel>	<p>The logging level for AP_GetCompanies.js.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
<CacheDuration>60</CacheDuration>	This element specifies the cache duration in minutes.
<MaxResults>255</ MaxResults >	This element specifies the maximum results to return from the staging table query. This value cannot exceed 255.
</AP_GetCompanies>	
<AP_GetData>	

Element	Description
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_GetData.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetData.js. 0 = least verbose 5 = most verbose The default is 4.
</AP_GetData>	
<AP_GetGLAcctCodes>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_GetData.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for GetGLAcctCodes.js. 0 = least verbose 5 = most verbose The default is 4.
<MaxResults type="int">255</MaxResults>	This element specifies the maximum constraint records to return at one time. This value cannot exceed 255.
</AP_GetGLAcctCodes>	
<AP_GetGLCodeList>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_GetGLAcctCodes.js to the console or to a log file. false = the console true = a log file The default is false.

Element	Description
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetGLCodeList.js. 0 = least verbose 5 = most verbose The default is 4.
<CacheEnabled type="boolean">true</CacheEnabled>	This element specifies whether to enable caching for this script.
<CacheDuration type="int">60</CacheDuration>	This element specifies, in minutes, how often to refresh the cache.
<PreCaches>	This element holds all of the caches generated by iScript.
<PreCache constraint1="ALL" constraint2="ALL" glnumber="1" forcerefresh="true"/>	This element is a single cache rule used to generate a cache when iScript runs.
</PreCaches>	
<MaxResults>1500</ MaxResults >	This element specifies the maximum GL codes to return at one time.
</AP_GetGLCodeList>	
<AP_GetGLCodeUserConstraints>	
<LogToFile type="boolean">>false</LogToFile>	This element sets the logging for AP_GetGLCodeUserConstraints.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetGLCodeUserConstraints.js. 0 = least verbose 5 = most verbose The default is 4.
<MaxResults>255</ MaxResults >	This element specifies the maximum constraint records to return at one time. This value cannot exceed 255.
</AP_GetGLCodeUserConstraints >	
<AP_GetPOs>	

Element	Description
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_GetPOs.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetPOs.js. 0 = least verbose 5 = most verbose The default is 4.
<MaxResults>255</ MaxResults >	This element specifies the maximum purchase orders to return at one time. This value cannot exceed 255.
</AP_GetPOs>	
<AP_GetPOLines>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_GetPOLines.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetPOLines.js. 0 = least verbose 5 = most verbose The default is 4.
<MaxResults>127</ MaxResults >	This element specifies the maximum PO lines to return at one time. This value cannot exceed 255.
</AP_GetPOLines>	
<AP_GetVendorLoc>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_GetPOLines.js to the console or to a log file. false = the console true = a log file The default is false.

Element	Description
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetVendorLoc.js. 0 = least verbose 5 = most verbose The default is 4.
<MaxResults>255</ MaxResults >	This element specifies the maximum vendor locations to return at one time. This value cannot exceed 255.
</AP_GetVendorLoc>	
<AP_GetVendors>	
<LogToFile type="boolean">>false</LogToFile>	This element sets the logging for AP_GetVendors.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_GetVendors.js. 0 = least verbose 5 = most verbose The default is 4.
<MaxResults>50</ MaxResults >	This element specifies the maximum vendor locations to return at one time. This value cannot exceed 255.
</AP_GetVendors>	
<AP_Invoice_Export>	
<LogToFile type="boolean">>false</LogToFile>	This element sets the logging for AP_Invoice_Export.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_Invoice_Export.js. 0 = least verbose 5 = most verbose The default is 4.
<ExportConfigFile>AP_Export.xml</ExportConfigFile>	This element specifies the configuration file to use for export mappings.

Element	Description
<DataFieldDelimiter> </ DataFieldDelimiter>	This element specifies the delimiter to use in the output file.
<QuotationWrapping>false </QuotationWrapping>	When set to true, this element specifies that all data fields are wrapped in quotations.
<AP520InvoiceFileName> AP520_INVOICE_%04d%02d%02d_%02d%02d%02d.csv </ AP520InvoiceFileName>	This element specifies the file name pattern to use for the Lawson-specific AP520 invoice file.
<AP520DistribFileName> AP520_DISTRIB_%04d%02d%02d_%02d%02d%02d.csv </ AP520DistribFileName>	This element specifies the file name pattern to use for the Lawson-specific AP520 distribution file.
<AP520CommentName> AP520_COMMENT_%04d%02d%02d_%02d%02d%02d.csv </ AP520CommentName>	This element specifies the file name pattern to use for the Lawson-specific AP520 comment file.
<MA540FileName> MA540_%04d%02d%02d_%02d%02d%02d.csv </ MA540FileName>	This element specifies the file name pattern to use for the Lawson-specific MA540 file.
<ExportSections>	The elements in this section flag whether to enable exports with the Lawson-specific AP_Export.xml file.  For example, to export header-level values from the Non-PO eForm to the AP520 Invoice file, under <NonPO>, set <Header type="boolean"> to true.
<NonPO>	This node corresponds to the <nonpo> section of the Lawson-specific AP_Export.xml file.
<Header type="boolean">true</Header>	Specifies whether to export non-PO header-level fields as a single header row in the AP520 Invoice file.  The setting is case-sensitive.
<GLLine type="boolean">true</GLLine>	Specifies whether to export each non-PO GL distribution line as a single detail row in the AP520 Distribution file.  The setting is case-sensitive.
<Comment type="boolean">true</Comment>	Specifies whether to export non-PO header comment data as a single comment row in the AP520 Comment file.  The setting is case-sensitive.
</NonPO>	
<PO>	This node corresponds to the <po> section of the Lawson-specific AP_Export.xml file.

Element	Description
<Header type="boolean">true</Header>	Specifies whether to export PO header-level fields as a single header row in the MA540 Invoice file. The setting is case-sensitive.
<POLine type="boolean">true</Comment>	Specifies whether to export each PO distribution line as a single detail row in the MA540 file. The setting is case-sensitive.
<GLLine type="boolean">false</GLLine>	Specifies whether to export each PO GL distribution, associated with a PO distribution line, as a single detail row in the MA540 file. The setting is case-sensitive.
<Misc type="boolean">true</Comment>	Specifies whether to export header-level miscellaneous data as a single row in the MA540 file. The setting is case-sensitive.
<Misc_Line type="boolean">false</Comment>	Specifies whether to export distributed fields individually, as miscellaneous lines with the Add On Cost (AOC) code, in the MA540 file. The setting is case-sensitive.
</PO>	
</ExportSections>	
<VoucherBuildFileName> VoucherBuild_%04d%02d%02d_%02d%02d%02d.csv </ VoucherBuildFileName>	This element specifies the File name pattern to use for the PeopleSoft Voucher Build file.
</AP_ Invoice_Export>	
<AP_Maintenance>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_Maintenance.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_Maintenance.js. 0 = least verbose 5 = most verbose The default is 4.
</AP_Maintenance>	

Element	Description
<AP_ReindexAndCPMapping>	
<LogToFile type="boolean">>false</LogToFile>	<p>This element sets the logging for AP_ReindexAndCPMapping.js to the console or to a log file.</p> <p>false = the console</p> <p>true = a log file</p> <p>The default is false.</p>
<DebugLevel type="int">4</DebugLevel>	<p>The logging level for AP_ReindexAndCPMapping.js.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
</AP_ReindexAndCPMapping>	
<AP_Validate_Server>	
<LogToFile type="boolean">>false</LogToFile>	<p>This element sets the logging for AP_Validate_Server.js to the console or to a log file.</p> <p>false = the console</p> <p>true = a log file</p> <p>The default is false.</p>
<DebugLevel type="int">4</DebugLevel>	<p>The logging level for AP_Validate_Server.js.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
<ValidateConfigFile> AP_Validate_Server.xml </ValidateConfigFile >	This element specifies the file that contains the configuration information for server validation.
<MaxResults>255</ MaxResults >	This element specifies the maximum results to retrieve at one time. This value cannot exceed 255.
</AP_Validate_Server>	
<AP_ValidateGLAcctCodes>	

Element	Description
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_ValidateGLAcctCodes.js to the console or to a log file.  false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_ValidateGLAcctCodes.js. 0 = least verbose 5 = most verbose The default is 4.
</AP_ValidateGLAcctCodes>	
<AP_ValidateGLCodes>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_ValidateGLCodes.js to the console or to a log file.  false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_ValidateGLCodes.js. 0 = least verbose 5 = most verbose The default is 4.
<CacheEnabled type="boolean">true</CacheEnabled>	This element specifies whether caching is enabled for this script.
</AP_ValidateGLCodes>	
<AP_VirtualTable_Installer>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_ValidateTable_Installer.js to the console or to a log file.  false = the console true = a log file The default is false.

Element	Description
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_VirtualTable_Installer.js. 0 = least verbose 5 = most verbose The default is 4.
</AP_VirtualTable_Installer>	
<AP_VirtualTable_Migrate>	
<LogToFile type="boolean">>false</LogToFile>	This element sets the logging for AP_VirtualTable_Migrate.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_VirtualTable_Migrate.js. 0 = least verbose 5 = most verbose The default is 4.
</AP_VirtualTable_Migrate>	
<AP_VirtualTable_MigratePurge>	
<LogToFile type="boolean">>false</LogToFile>	This element sets the logging for AP_VirtualTable_MigratePurge.js to the console or to a log file. false = the console true = a log file The default is false.
<DebugLevel type="int">4</DebugLevel>	The logging level for AP_VirtualTable_Migrate.js. 0 = least verbose 5 = most verbose The default is 4.
</AP_VirtualTable_MigratePurge>	
<AP_VirtualTable_Updater>	

Element	Description
<code>&lt;LogToFile type="boolean"&gt;false&lt;/LogToFile&gt;</code>	<p>This element sets the logging for AP_VirtualTable_Updater.js to the console or to a log file.</p> <p>false = the console</p> <p>true = a log file</p> <p>The default is false.</p>
<code>&lt;DebugLevel type="int"&gt;4&lt;/DebugLevel&gt;</code>	<p>The logging level for AP_VirtualTable_Updater.js.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
<code>&lt;UpdaterConfigFile&gt;AP_VirtualTable_Updater.xml&lt;/ UpdaterConfigFile &gt;</code>	<p>This element specifies the file that contains the configuration information for updating virtual tables.</p>
<code>&lt;DataFieldDelimiter&gt;,&lt;/ DataFieldDelimiter &gt;</code>	<p>This element specifies the delimiter to use in the input files to the VirtualTable_Updater.</p>
<code>&lt;Connector&gt;CSV&lt;/Connector&gt;</code>	<p>This element specifies which connector to use to gather the input for updating the virtual tables.</p>
<code>&lt;AppendDateTimeStampToFile&gt;true&lt;/AppendDateTimeStampToFile&gt;</code>	<p>This element specifies whether the process updates the CSV file name by appending <code>_YYYYMMDD_hhmmss</code> to the file name.</p>
<code>&lt;/AP_VirtualTable_Updater&gt;</code>	
<code>&lt;/scripts&gt;</code>	
<code>&lt;/configuration&gt;</code>	

## AP\_Export.xml

This file is stored in the \inserver6\etc\AP directory. The elements in this file vary depending on the business application used for the implementation.

The output date format is configurable in AP\_Export.xml. Valid formats are MMDDYYYY, DDMMYYYY, and YYYYMMDD (slashes are optional). If it is not specified, the default format is MMDDYYYY.

Element	Description
<configuration errorqueue="AP Error" stampexported="true" update_po_table="true" routeforward="true" nextqueue="" outputdir="C:/inserver6/temp/ap/export">	
errorqueue=" <i>queue_name</i> "	<p>The errorqueue attribute of the &lt;configuration&gt; element must contain the name of the workflow queue where workflow items are placed when unexpected errors occur. This name must exactly match a queue name in the workflow process in ImageNow.</p> <p>This errorqueue can be a different queue than the queue used in the AP_Skin.xml file. After the data is exported, it sets the stampexported attribute to true.</p>
stampexported="true"	<p>Set the stampexported attribute of the &lt;configuration&gt; element to true if you want forms updated with a time and date stamp when the data is exported to your business application. The AP_Skin.xml file uses this element to prevent users from modifying a previously exported form when the value of the &lt;readonlyifexported&gt; element is set to true.</p> <p>If the workflow item has been exported before, it is sent to the queue specified in the errorqueue attribute instead.</p>
update_po_table="true"	<p>Set this element to true if you are using virtual tables and you want to update the time and date stamp so that the invoice is seen as previously exported. Use this if you want the invoice to be used later for purge criteria.</p>
writeheader="true"	<p>This attribute requires a value of true or false. Set it to true to write out a header row with column headings. Set it to false otherwise. This value should be set to true for most implementations.</p> <p>Set this attribute to true if you want the header at the first line of the CSV file to contain the field names. If not, set it to false.</p>
routeforward="true"	<p>This attribute requires a value of true or false. Set the value to true if the iScript (AP_Invoice_Export.js) should route the workflow item forward. Set the value to false if you want the user to select the next queue for routing. If set to true, the route on the export queue must be of the Sequential-Auto type.</p> <p>If you set this attribute to false, you must specify a value for the nextqueue attribute of the configuration element.</p> <p>You can set the export queue to be executed once per day by using an iScript set to execute from INTTool.</p>

Element	Description
<code>nextqueue="queue_name"</code>	This attribute holds the name of the workflow queue to which the iScript (AP_Invoice_Export.js) should route the workflow item after processing is complete. Set the <code>routeforward</code> attribute to false if using a queue for this attribute.
<code>outputdir="C:/inserver6/temp/ap/export"</code>	<p>This attribute holds the file path of the location where exported files are placed after processing is complete. Configure this path if your ImageNow Server is not on the C:\ drive, if you use non-Windows platform, or have an active-active server environment.</p> <p>If you use forward slashes, the path works on Windows and non-Windows platforms. Do not use a trailing slash at the end of the path.</p>
<code>&lt;queues&gt;</code>	You can create a <code>&lt;queue&gt;</code> element for each workflow queue you need to export data from in this <code>&lt;queues&gt;</code> element. The value for the <code>&lt;queue&gt;</code> element is the name of the workflow queue where the invoices to upload are located. The APO Invoice Upload queue value is set in this file by default when installed. However, you need to create this queue (and any other queues you add to this element) in ImageNow if the queues do not already exist.
<code>&lt;queue&gt;queue_name&lt;/queue&gt;</code>	This element holds the name of the workflow queue that holds the invoices waiting uploading to the business application. If there is more than one queue, add an element for each one.
<code>&lt;/queues&gt;</code>	
<code>&lt;worksheets&gt;</code>	This element holds the name of the AP Invoice eForm. This value is set by default when installed. For most implementations, there should be only one form, and the default name of 'AP Invoice' should be unchanged.
<code>&lt;worksheet&gt;AP Invoice&lt;/worksheet&gt;</code>	
<code>&lt;/worksheets&gt;</code>	
<code>&lt;invoice&gt;</code>	This element holds the data mapping configuration elements for form data. These elements map the form data fields to the export file fields. Typically, these values do not need to be modified. However, if you used optional fields in your form, you configure those fields here for exporting.
<code>&lt;header&gt;</code>	

Element	Description
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</header>	
<poline>	
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</poline>	
<glline>	

Element	Description
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</glline>	
</invoice>	
<nonpo>	<p>This element holds the data mapping configuration elements for form data. These elements map the form data fields to the export file data for non-PO invoices. Typically, these values do not need to be modified. However, if you used optional fields in your form, you configure those fields here for exporting.</p>
<header>	

Element	Description
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</header>	
<glline>	
</glline>	
</nonpo>	
<po>	<p>This element holds the data mapping configuration elements for the form data. These elements map the form data fields to the export file data for PO invoices. Typically, these values do not need to be modified. However, if you used optional fields in your form, you configure those fields here for exporting.</p>
<header>	

Element	Description
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</header>	
<poline>	
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</poline>	
<glline>	

Element	Description
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</glline>	
<misc>	
<field></field>	<p>The required <b>name</b> attribute must be unique within the node.</p> <p>The optional <b>sequence</b> attribute is only for your reference or documentation purposes.</p> <p>The <b>worksheet</b> attribute ties the &lt;field&gt; element to a field on the eForm.</p> <p>The <b>type</b> attribute sets the value type. Options include alpha, numeric, date, and decimal,</p> <p>The <b>format</b> attribute sets the date format for the type attribute, if set to date. You may use any combination of dd, mm, yy and yyyy, with or without slashes. For example, ddmmyy or dd/mm/yyyy.</p> <p>The <b>constant</b> attribute controls constant variables.</p> <p>The <b>length</b> attribute sets the numeric limit of the field. The eForm provides validation on this limit in the user interface.</p> <p>The <b>whole</b> attribute allows you to specify the number of characters before the decimal, and the <b>decimal</b> attribute allows you to specify the number of characters after the decimal. The whole and decimal attributes combined should equal the length attribute.</p>
</misc>	
<po>	
</configuration>	

## AP\_Validate\_Server.xml

This file is stored in the [drive:]\\inserver6\\etc\\AP directory.

Element	Description
<pre>&lt;configuration errorqueue="AP Error" worksheet="AP Invoice" pomatching="true" routeback="true"&gt;</pre>	<p>This element handles validation errors that may occur during use of the eForm.</p> <p><b>errorqueue.</b> The name of the workflow queue where workflow items route when an unexpected validation error occurs.</p> <p><b>worksheet.</b> The name of the AP Invoice eForm. The default value is "AP Invoice."</p> <p><b>pomatching.</b> When set to true, imports PO header-line information from the business application or virtual tables. If Intelligent Capture captures this data, set this value to false. This attribute does not apply to non-PO invoices.</p> <p><b>routeback.</b> When set to true, routes the document to the previous workflow queue if a validation error occurs in the AP_Validate_Server.js script.</p>
<datamappings>	
<pre>  &lt;datamapping enabled="true" variance_calc="true" doctype="Invoice"&gt;</pre>	<p>This element synchronizes eForm values with document keys and custom properties.</p> <p><b>enabled.</b> If set to true, enables the synchronization between eForm fields and ImageNow document properties.</p> <p><b>variance_calc.</b> If the solution processes PO-based invoices and the workflow routing rules use variance calculations, set to true.</p> <p><b>doctype.</b> Indicates whether the datamapping node configures the Invoice or Credit Memo document type.</p>
<pre>    &lt;customprops enabled="true"&gt;       &lt;customprop/&gt;       &lt;customprop/&gt;     &lt;/customprops&gt;</pre>	<p>The installer sets the custom property elements by default when you install the AP Invoice eForm. Modify the custom property name, the eForm field name, and the enabled attribute as needed.</p>
<pre>    &lt;dockeys enabled="true"&gt;       &lt;dockey/&gt;       &lt;dockey/&gt;     &lt;/dockeys&gt;</pre>	<p>The installer sets the document property elements by default when you install the AP Invoice eForm. Modify the custom property name, the eForm field name, and the enabled attribute as needed.</p>
</datamapping>	
</datamappings>	

Element	Description
<code>&lt;required_fields enabled="true"&gt;</code>	<p>This element identifies and configures server validation of the AP Invoice eForm fields that require user input. The installer sets these values with defaults during installation.</p> <p>Typically, you do not need to modify the default values.</p> <p><b>enabled.</b> If set to true, allows validation in the following enabled nodes and fields. The default is true.</p>
<code>&lt;pohdr ponumber="PONumber" enabled="true"&gt;</code>	<p>This node enables server validation of the invoice header fields for purchase order-based invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is true.</p>
<code>&lt;worksheet enabled="true" field="[AP Invoice eForm field]" type="[type]"&gt;&lt;/worksheet&gt;</code>	<p>These elements enable server validation of the following PO-based invoice fields.</p> <ul style="list-style-type: none"> <li>• CompanyNumHdr</li> <li>• PONumber</li> <li>• InvoiceNumber</li> <li>• InvoiceDate</li> <li>• TotalAmount</li> </ul> <p><b>enabled.</b> If set to true, allows validation of this element. The default for each element is false.</p> <p><b>type.</b> Defines the type, when necessary. Supported types are date, number, and boolean. The default for undefined types is string.</p>
<code>&lt;/pohdr&gt;</code>	
<code>&lt;nonpohdr enabled="true"&gt;</code>	<p>This node enables server validation of the invoice header fields for non-PO invoices</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements.</p>

Element	Description
<worksheet enabled="true" field=[AP Invoice eForm field]></worksheet>	<p>These elements enable server validation of the following non-PO fields.</p> <ul style="list-style-type: none"> <li>• CompanyNumHdr</li> <li>• InvoiceNumber</li> <li>• InvoiceDate</li> <li>• TotalAmount</li> <li>• VendorID</li> </ul> <p><b>enabled.</b> If set to true, allows validation of this element. The default for each element is false.</p> <p><b>type.</b> Defines the type, when necessary. Supported types are date, number, and boolean. The default for undefined types is string.</p>
</nonpohdr>	
<creditdr enabled="true">	<p>This node enables server validation of the invoice header fields for credit-based invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements.</p>
<worksheet enabled="true" field=[AP Invoice eForm field]></worksheet>	<p>These elements enable server validation of the following credit-based invoice fields.</p> <ul style="list-style-type: none"> <li>• CompanyNumHdr</li> <li>• VendorID</li> <li>• InvoiceNumber</li> <li>• InvoiceDate</li> <li>• TotalAmount</li> </ul> <p><b>enabled.</b> If set to true, allows validation of this element. The default for each element is false.</p> <p><b>type.</b> Defines the type, when necessary. Supported types are date, number, and boolean. The default for undefined types is string.</p>
</creditdr>	
<podetail enabled="false">	<p>This node enables server validation of the invoice detail fields for PO-based invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is false.</p>

Element	Description
<code>&lt;worksheet enabled="false" field=[AP Invoice eForm field]&gt;&lt;/worksheet&gt;</code>	<p>These elements enable server validation of the following PO-based invoice fields.</p> <ul style="list-style-type: none"> <li>• LineNumber</li> <li>• ItemNumber</li> <li>• ItemDescription</li> <li>• Quantity</li> <li>• UOM</li> <li>• ExtendedAmount</li> <li>• POLVATJurisdiction</li> <li>• POLVATCode</li> <li>• POLVATAmount</li> <li>• PO1 through PO10</li> </ul> <p><b>enabled.</b> If set to true, allows validation of this element. The default for each element is false.</p> <p><b>type.</b> Defines the type, when necessary. Supported types are date, number, and boolean. The default for undefined types is string.</p>
<code>&lt;/podetail&gt;</code>	
<code>&lt;glDetail enabled="false"&gt;</code>	<p>This node enables server validation of the invoice detail fields for GL-based invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is false.</p>

Element	Description
<code>&lt;worksheet enabled="false" field=[AP Invoice eForm field]&gt;&lt;/worksheet&gt;</code>	<p>These elements enable server validation of the following GL-based invoice fields.</p> <ul style="list-style-type: none"> <li>• LinePCT</li> <li>• LineAmount</li> <li>• GL1 through GL24</li> <li>• CompanyNum</li> <li>• GLUnit</li> <li>• GLAcct</li> <li>• GLAcctDesc</li> <li>• GLAcct5</li> <li>• GLAcct6</li> <li>• GLVATJurisdiction</li> <li>• GLVATCode</li> <li>• GLVATAmount</li> </ul> <p>The default for each element is false.</p>
<code>&lt;/gldetail&gt;</code>	
<code>&lt;/required_fields&gt;</code>	
<code>&lt;glcode_validation enabled="false"&gt;</code>	<p>GL Code validation runs validation logic for each enabled GL Code field. The server validates each GL Code field individually against the records in the virtual tables.</p> <p>GL Codes are a part of the GL Lines in non-PO and Credit invoices, and in the Additional Entries section of PO invoices. This validation ignores user constraints.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is false.</p>
<code>&lt;worksheet enabled="false" field=[AP Invoice eForm field]&gt;&lt;/worksheet&gt;</code>	<p>These elements enable server validation of the following invoice fields.</p> <ul style="list-style-type: none"> <li>• GL1 through GL24</li> </ul> <p>The default for each element is false.</p>
<code>&lt;/glcode_validation&gt;</code>	

Element	Description
<glaccount_validation enabled="false">	<p>GL Account validation runs validation logic for each enabled GL Account field. The server validates enabled fields, as a group, against the virtual tables.</p> <p>GL Accounts are a part of the GL Lines in non-PO and Credit invoices, and in the Additional Entries section of PO invoices.</p> <p>The validation logic indicates GL Account fields are missing if all of the enabled fields are blank. The validation indicates the fields are invalid if any of the enabled fields are blank or contain invalid values.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is false.</p>
<worksheet enabled="false" field=[AP Invoice eForm field]></worksheet>	<p>These elements enable server validation of the following invoice fields.</p> <ul style="list-style-type: none"> <li>• CompanyNum</li> <li>• GLUnit</li> <li>• GLAcct</li> <li>• SubAcct</li> <li>• GLAcct5</li> <li>• GLAcct6</li> </ul> <p>The default for each element is false.</p>
</glaccount_validation.	
<po_distribution_validation enabled="true">	<p>This node validates the Total Amount distribution and any additional amount distributions in PO-based invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is true.</p>
<include[AP Invoice eForm amount field] enabled="false" distributed="false">	<p>If enabled, these elements include the amount in the distribution calculation. This node includes the following amount fields.</p> <ul style="list-style-type: none"> <li>• tax</li> <li>• freight</li> <li>• misc</li> <li>• addl_amt1 through addl_amt8</li> </ul> <p><b>distributed.</b> If set to true, enables validation of the specified header amount distribution.</p> <p>The default for each element is false.</p>
</po_distribution_validation>	

Element	Description
<nonpo_distribution_validation enabled="true">	<p>This node validates the Total Amount distribution in non-PO invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is true.</p>
<include[AP Invoice eForm amount field] enabled="false">	<p>If enabled, these elements include the amount in the distribution calculation. This node includes the following amount fields.</p> <ul style="list-style-type: none"> <li>• tax</li> <li>• freight</li> <li>• misc</li> <li>• addl_amt1 through addl_amt8</li> </ul> <p>The default for each element is false.</p>
</nonpo_distribution_validation>	
<credit_distribution_validation enabled="true">	<p>This node validates the Total Amount distribution in credit invoices.</p> <p><b>enabled.</b> If set to true, allows validation of the following enabled elements. The default is true.</p>
<include[AP Invoice eForm amount field] enabled="false">	<p>If enabled, these elements include the amount in the distribution calculation. This node includes the following amount fields.</p> <ul style="list-style-type: none"> <li>• tax</li> <li>• freight</li> <li>• misc</li> <li>• addl_amt1 through addl_amt8</li> </ul> <p>The default for each element is false.</p>
</credit_distribution_validation>	
<vatsummary_validation enabled="false"/>	<p>The vatsummary_validation attribute controls whether to complete a server-side validation of the header-level VAT Total against the line-level VAT sum. For more information, refer to the <a href="#">Configure the VAT elements in the XML files</a> section.</p>
</configuration>	

## AP\_Custom\_Cultures.xml

This file is stored in the \inserver6\etc\ap directory.

**Note** The installer creates this file if it does not already exist locally. In an upgrade scenario, the installer does not merge or overwrite this file.

Element	Description
<cultures>	
<code>&lt;culture name="unique_name" base="language code-country/region code" language="language code"&gt;</code>	<p>The culture element allows you to customize your form according to your locale.</p> <p>The name attribute specifies the unique name for the culture in the form.</p> <p>The base attribute specifies the language code from which culture inherits properties. The base attribute consists of the two-digit language code or the two-digit language code and the two-digit country or region code and is formatted as follows:</p> <pre>&lt;language code&gt; &lt;language code&gt;-&lt;country/region code&gt;</pre> <p>The default is en-US.</p> <p>The language attribute specifies the top level, more general, neutral culture. The language attribute consists of the two-digit language code.</p>
<code>&lt;numberformat groupseparator="separator_value" decimalseparator="separator_value" groupsizes="groupsize_values"&gt;</code>	<p>The numberformat element specifies formatting rules for general number values.</p> <p>The groupseparator attribute specifies the separator character between different groups of digits.</p> <p>The decimalseparator attribute specifies the separator character between the integer part of the number and the decimal part.</p> <p>The groupsizes attribute specifies how many digits are included in each group of the number.</p>
</numberformat>	
<code>&lt;calendar separator="separator_value"&gt;</code>	<p>The calendar element specifies formatting rules for date values.</p> <p>The separator attribute specifies the separator character between the day, month, and year values of the date string.</p>

Element	Description
<code>&lt;patterns shortinput="short_date" longinput="long_date" display="date"&gt;</code>	<p>The patterns element specifies formatting rules for date values.</p> <p>The shortinput attribute specifies the input short date format, without separators, such as MMddyy.</p> <p>The longinput attribute specifies the input long date format, without separators, such as MMddyyyy.</p> <p>The display attribute specifies the display date format, such as MM/dd/yy.</p> <p><b>Note</b> The separator value for the display date should match the separator attribute value you specify in the calendar element.</p>
<code>&lt;/calendar&gt;</code>	
<code>&lt;messages&gt;</code>	
<code>&lt;message key="caption key" caption="caption"&gt;</code>	<p>The message element allows you to configure eForm label captions, error messages, and tooltips.</p> <p>The key attribute specifies an eForm caption key.</p> <p>The caption attribute specifies the label caption, error message, or tooltip value that you want to display for that field.</p> <p><b>Note</b> For a complete list of the available eForm caption keys, refer to <a href="#">Appendix B: Caption keys and captions</a>.</p>
<code>&lt;/messages &gt;</code>	
<code>&lt;/culture&gt;</code>	
<code>&lt;/cultures&gt;</code>	

## AP\_VirtualTable\_Updater.xml

This file is stored in the \inserver6\etc\AP directory.

Element	Description
<pre>&lt;configuration inputpath="C:\inserver6\temp\ap\import\*.csv" containsheaderline="true" archivepath="C:\inserver6\temp\ap\archive" errorpath="C:\inserver6\temp\ap\error"&gt;</pre>	<p>The host attribute is set to your business application by default when you install the form. Configure these paths if your ImageNow Server is not on the C:\ drive, you use a non-Windows platform, or you have an active-active server environment.</p>
<pre>  &lt;recordstructure recordtype="&lt;virtual table alias&gt;" table="&lt;folder type name&gt;" enabled="true"&gt;</pre>	<p>folder type name = virtual table name</p>
<pre>    &lt;column customprop="&lt;virtual table column" enabled="true"&gt;&lt;/column&gt;</pre>	<p>These elements contain default initial values.</p> <p>You do not need to modify these values unless you use a different structure in your CSV files. For example, enabling columns for VAT. The structure in this file needs to match the structure in your data files.</p> <p>There are several virtual tables and a corresponding &lt;recordstructure&gt; element for each virtual table. Some of the virtual tables are BUS_UNIT, GL_ACCT, PO, PO_LINE, and VENDOR.</p>
<pre>  &lt;/recordstructure&gt;</pre>	
<pre>&lt;/configuration&gt;</pre>	

## Configure workflow for AP Invoice eForm

After you run the AP Invoice eForm installation wizard, you must configure the form to work with your existing AP workflow process.

If you do not have an error queue defined in your workflow process, you must add one. Workflow uses this queue to remove invoices with errors other than simple validation errors from the AP workflow process.

You must add the AP Invoice eForm to every workflow queue that plays one of the following roles:

- Invoice Entry
- Invoice Processing
- Invoice Approval

### About the reindex document and custom property mapping script

The AP\_ReindexAndCPMapping.js script synchronizes form values to custom properties and document keys based on the data mapping parameters in AP\_Validate\_Server.xml.

### About the validation script

The following procedure applies to each queue that is assigned any of the following roles for the form:

- Invoice Entry
- Invoice Processing
- Invoice Approval

You must associate the AP\_Validate\_Server.js script with the next queue in the process, preferably a System queue, which also manages the routing of the item.

#### Non-PO invoices

For non-PO invoices, this script synchronizes form values to custom properties and document keys based on the data mapping parameters in AP\_Validate\_Server.xml.

#### PO-based invoices

This script calculates a variance between the actual amount of the invoice and the expected invoice amount based on the quantity received but not invoiced. The script calculates the difference as a variance amount and a variance factor. Then, the script loads the difference as a custom property for the Invoice document type. You can use a routing rule to use the variance amount, variance factor, or a combination of the two to route the invoice accordingly.

For example, if the script determines that the invoice falls within the variance amount or factor, it routes the invoice to the next workflow queue. If it falls outside the variance amount or factor, it routes the invoice to a matching exception queue for manual attention.

## About workflow queue roles

Each workflow queue where the eForm is used must be listed in the <roles> section of the AP\_Skin.xml file in the \inserver6\etc\ap directory. This file determines what role the eForm plays in the workflow queue and how to view the form. When determining the role for a queue, ImageNow first checks under the <roles> section of the AP\_Skin.xml file for a queue or sub queue name. If ImageNow finds a queue or sub-queue name, the <role name> value associated with that queue or sub-queue determines the role. If ImageNow does not find a queue or sub queue name, it then checks for a super queue name. If ImageNow finds a super queue name, the <role name> value associated with that super queue determines the role. All sub queues of the super queue also inherit that role. If ImageNow does not find any queue names in the XML file, it uses the role of (default).

For detailed information about the settings in this file, refer to the [AP\\_Skin.xml](#) table in the [AP Invoice eForm configuration files](#) section of this document.

The following table outlines the relationship between the queue role, queue name, and user roles.

Role name	Role description	User role
Invoice Entry	Queues with this role are used for incoming invoices and header data entry.	Use this role for initial invoice entry (header data) or for invoice detail data if needed. Typically, this role is configured to allow entry into invoice header only. Detail data is usually disabled (not displayed).
Invoice Processing	Queues with this role are used to capture and validate GL coding, match PO invoices, and route to approval.	Use this role for GL Coding Entry (non-PO Invoices) and Line Item Matching (PO Invoices). Typically, this role is configured to allow entry or modification to both the invoice header and detail.
Invoice Approval	Queues with this role are used to approve invoices after processing.	Use this role for invoice approval. Typically, this role is configured to allow review of the invoice header (no changes), as well as entry or modification to the invoice detail.
Default	Queues with this role are used for accessing the form outside of a workflow queue or for test purposes.	With this role, the form is read-only. This default role for users and groups who need access from workflow queues that are not defined in the other roles. You can also use this role for users or groups who need access to the form outside of workflow, such as the Documents and Projects views accessed from the Views pane.

The invoice header data, such as invoice amount or date, is the same for PO and non-PO invoices. For PO invoices, the detail information includes line items from the purchase order used for matching purposes and potentially GL coding entries for header amount fields that need distributed. For non-PO invoices, the detail information includes GL coding entries.

## Reserved characters in XML and forms

**Note** Using special characters in queue names is not recommended.

In XML, some characters are reserved because these characters are markup delimiters. Therefore, these reserved characters can never appear as a literal string in XML character data (such as the text value of an element). Literal string means a letter or symbol that stands for itself as opposed to a feature, function, or entity associated with it in a programming language. For example, \$ can be a symbol that refers to the end of a line, but as a literal, it is a dollar sign. Exceptions to this rule are when these characters are used as markup delimiters, in comments, in processing instructions, and in CDATA sections.

Sometimes it is necessary to display XML reserved characters as text. In order to do this, you must use an escape sequence. An escape sequence is a sequence of special characters that sends a command to a device or program. Typically, an escape sequence begins with an escape character, but this is not universally true. XML provides standard escape sequences, called character entity references, for these reserved characters. These entity references provide a way to refer to a character that is not universally encodable.

The following table illustrates these reserved characters and their replacement entity references.

Reserved Character	Meaning	Entity Reference/Escape Sequence	Comments/What to use when coding a form
>	Greater than	&gt;	For compatibility, this character must be escaped using the entity reference or as a character reference when it appears in the string "]]>" in content and when that string is not marking the end of a CDATA section.
<	Less than	&lt;	This character should always appear in content as an escape character so that it does not interact with the syntax of the markup.
&	Ampersand	&amp;	This character should always appear in content as an escape character so that it does not interact with the syntax of the markup.
'	Apostrophe	&apos;	You can symbolize the apostrophe or single-quote character (') with this character entity reference when you need to embed a single-quote or apostrophe inside a string that is already single-quoted.  Although &apos; is part of the XML language, it is not defined in HTML. Use &#39; if text is passed to an HTML user agent.
"	Quotation mark	&quot;	You can symbolize the double-quote character (") with this character entity reference when you need to embed a double-quote inside a string that is already double-quoted.  <b>Note</b> Use \" when writing unparsed data to XML.
%	Percent	&#37;	The Notification Services XML vocabulary reserves the percent sign (%) for denoting parameters.  <b>Note</b> \$#37 is an escape sequence, but not an entity reference defined by XML.
\	Slash	None required	<b>Note</b> Use \\ when writing unparsed data to XML.

## Custom Logic

### Custom functions

You can use the following functions to customize your AP Invoice eForm.

Function	Description
custom_headerPermission	Set this function to true to enable header customization.
custom_detailPermission	Set this function to true to enable detail header customization.
custom_glDetailPermission	Set this function to true to enable GL detail header customization.
custom_glDetailPOPermission	Set this function to true to enable GL detail PO invoice header customization.
custom_glDetailNonPOPermission	Set this function to true to enable GL detail Non-PO invoice header customization.

### Global variables

These variables are available for use in the AP\_CustomValidation\_Client.js for custom scripting.

Variable	Description
gInitialTab	The value of the document's invoice type. For example, PO, non-PO or credit memo.
gUserName	The value of the user currently viewing the eForm.
gDocID	The value of the document ID.
gExportDate	The date that AP_Invoice_Export.js exported the document.
gDocType	The document's type.
gFolder	The value of the document's folder index key.
gTab	The value of the document's tab index key.
gField1	The value of the document's Field1 document property.
gField2	The value of the document's Field2 document property.
gField3	The value of the document's Field3 document property.
gField4	The value of the document's Field4 document property.

Variable	Description
gField5	The value of the document's Field5 document property.
gDrawer	The value of the document's drawer.
gQueue	The value of the queue where the document exists at the time the script runs.

## Add custom business logic to the form

Use the following methods, called user exits, to add your own business logic to AP Invoice eForm without causing the solution to fail.

- Adding JavaScript event handlers to form field events such as onclick and onblur.
- Calling a server-side iScript from the form to complete database lookups or validation.
- Supplementing the validation script provided with AP Invoice eForm (the inbound iScript that runs after an invoice is routed forward).

The following sections explain these user exits.

## JavaScript event handlers

The file AP\_CustomValidation\_Client.js has user exit points where you can add custom logic to handle various events that occur in the AP Invoice eForm. The following custom scripting events are available:

- When the user opens the form (body onload event).
- When the user changes a field's value in a form. You can perform event handling, such as onclick or onblur.
- When you use custom code to handle the following JavaScript events.

Event	Description
onblur	Occurs when an element loses focus.
onclick	Occurs when the user clicks an object.
onfocus	Occurs when an element has focus.
onkeydown	Occurs when a keyboard key is pressed.
onkeypress	Occurs when a keyboard key is pressed or held down.
onkeyup	Occurs when a keyboard key is released.
onmouseover	Occurs when the mouse button is moved over an element.
onmouseout	Occurs when the mouse button is moved off an element.

Event	Description
onmouseover	Occurs when the mouse button is released.

- When the user clicks the Validate button, either before the validate event or after the validate event, the following actions occur:
  1. The before validate event runs. You can insert custom logic before out-of-the box validation code runs.
  2. Standard out-of-the-box validation code runs.
  3. The after validate event runs. You can insert custom logic after the out-of-the box validation code runs.
- When the user adds a row to the eForm.
- When the user closes the form (body onunload event).

## Call a server-side iScript

You can use the iScript dbcall method to call your custom iScript. For example, you can call your own iScript file with the dbcall method when an event such as button click occurs in a form field. A generic button is available for this purpose. For more information on using the dbcall method, refer to the "Form iScript attributes and parameters" topic in ImageNow *Administrator Help*.

An iScript named AP\_GetData\_Custom.js provides the basic mechanism for exchanging data between the form and your external data sources.

The following example shows how you might modify the AP\_GetData\_Custom.js file.

```
function main ()
{
    debug = new iScriptDebug("USE SCRIPT FILE NAME", LOG_TO_FILE,
        DEBUG_LEVEL);

    debug.showINowInfo("INFO");

    try
    {
        if( ! CONFIG_VERIFIED )
        {
            var errorStr = "Configuration not verified. Please verify \n" +
                "the defines in the *** Configuration *** section at the\n" +
                "top \n" +
                "of this script and set CONFIG_VERIFIED to true.\n" +
                "Aborting.\n\n";
            debug.log("CRITICAL", errorStr);
            INprintf(errorStr);

            setOutputParams( new Array( "customData.errorMsg = '" +
                "AP_GetData_Custom.js(): Configuration not verified.\n" +
                "' + ';' ) );
            return;
        }

        var param = getInputParams();

        if (param.length>0)
```

```

{
    var arg1 = trim(param[0]);
    debug.log("INFO","Input Argument 1: %s\n", arg1);
}

//custom code goes here
var rtnJson = "customData.success = true";

debug.log("DEBUG", "Returning: %s\n", rtnJson);
setOutputParams( new Array( rtnJson ) );

```

## Enhance the validation iScript

You can use an inbound script to route an item forward to the next queue by writing an iScript file and then calling that file from the AP\_Validate\_Server.js file. You can also insert the custom script directly in the AP\_Validate\_Server.js file. The AP\_Validate\_Server.js file calls your iScript file and then performs that event. The AP\_Custom\_Validate\_Server.js file is available for you to write your custom code. It is called by AP\_Validate\_Server.js to run your validation.

The following example shows how you might modify the AP\_Custom\_Validate\_Server.js file.

```

function custom_validation(wsXML)
{
    debug.log("INFO", "\n");

    debug.log("INFO", "Custom Validation:\n");

    if (wsXML==null)
    {
        debug.log("ERROR", "\twsXML object is null.\n");
        return;
    }

    //insert your custom logic here

    //examples of how to traverse the wsXML object
    try
    {
        debug.log("INFO", "\tInvoice Number: %s\n", wsXML.InvoiceNumber);
        debug.log("INFO", "\tVendor ID: %s\n", wsXML.VendorID);
        //add your custom validation logic here

    }

    catch (e)
    {
        return 'Error: custom_validation()' + e.toString();
    }

    //use next line if you want to dump contents of wsXML object
    //return "Custom Validation Error";

    //use next line to return no error
    return "";
}

```

## Enhance the export iScript

The following section provides examples that you can use to enhance the export iScript.

### Field level sample

In the following example, the field export massager passes the function two parameters, the field name and the field value (by reference).

```
function custom_field_massager(fieldName, &fieldValue)
{
    debug.log("DEBUG", "Entering function custom_field_massager() ...\n");

    debug.log("DEBUG", "custom_massager -- fieldName:[%s], fieldValue:[%s] --\n",
fieldName, fieldValue);

    //IF FIELD IS MA-VENDOR, prepend 0 padding to 10 characters
    if(fieldName == "MA-VENDOR")
    {
        while(fieldValue.length < 10)
        {
            fieldValue = "0"+fieldValue;
        }
    }
    debug.log("DEBUG", "custom_field_massager() ... returning fieldValue:[%s]\n",
fieldValue);
    debug.log("DEBUG", "Exiting function custom_field_massager() ...\n");
}
```

### Global level sample

In the following example, you can customize the global export massager to configure an object to represent all of the data you want to export. For more examples, see the AP\_Custom\_Invoice\_Export.js.

```
function custom_export_massager(&exportObj)
{
    debug.log("DEBUG", "Entering function custom_export_massager() ...\n");
    //MODIFY CONTENTS OF THE HEADER DATA
    //EXAMPLE: IF Vendor is '3004' then set Location and Disc-Date to some value
    if(exportObj.header)
    {
        if(exportObj.header['MA-VENDOR'] == "3004")
        {
            exportObj.header['MA-LOCATION'] = "test location";
            exportObj.header['MA-DISC-DATE'] = "12/31/2010";
        }
    }
    debug.log("DEBUG", "Exiting function custom_export_massager() ...\n");
}
```

## Demonstration setup of AP Invoice eForm

You can add the form to queues in your AP workflow process and use the following procedures to set up the environment for the form. In addition, you can use the sample workflow setup information below to assist you with the tailoring or creation of your AP invoice workflow process.

### Load Data

Before building the workflow process, load the data into the virtual tables. Your installation directory includes a sample set of CSV files that can be used. Use the following instructions to load the data:

1. On the ImageNow Server computer, place your CSV files in the `\inserver6\temp\ap\import` directory. Note that you can modify this path by changing the value of the appropriate attribute in the XML configuration files. Refer to the configuration section of this document for more information.
2. Start INTTool by accessing a command prompt and changing directories to `\inserver6\bin`.
3. Enter the following command.

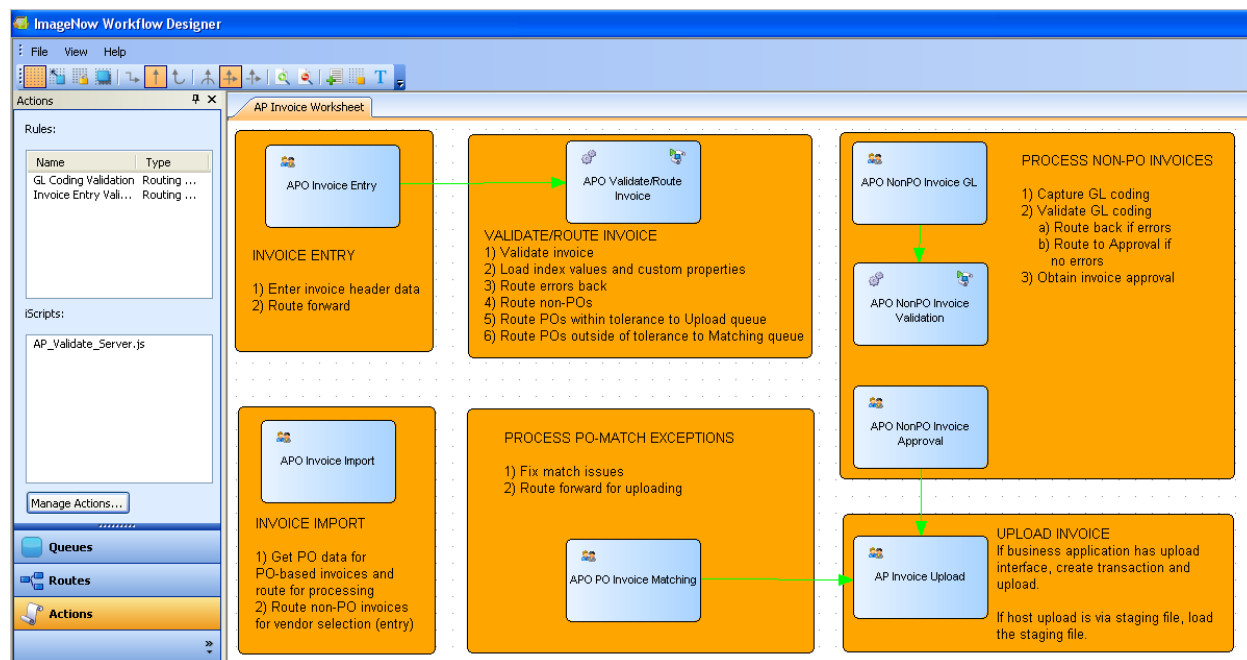
```
intool --cmd run-iscript --file AP_VirtualTable_Updater.js
```

4. After the command completes, complete the following substeps to verify that it was successful:
  1. Navigate to the `\inserver6\log` directory and open the `AP_VirtualTable_Updater_[date].log` file in a text editor.
  2. Verify that the following lines appear at the end of the file:

```
[current date] [current timestamp] AP_VirtualTable_Updater:00 [notify]
DONE
[current date] [current timestamp] AP_VirtualTable_Updater:00 [info]
Script run time: [seconds]
```

## Create AP workflow process

The following is a sample workflow process:



The example procedures in this section address workflow process setup only as it applies to the implementation of the AP Invoice eForm solution. For more information on completing other workflow tasks (such as designating workflow queue leads or drawing containers), refer to the *ImageNow Workflow Designer Help* available from the Perceptive Software Customer Support Portal.

You are not required to use the workflow queue names listed in this procedure, but ensure that you update the <roles> element in the AP\_Skin.xml file to reflect your queue names. Refer to the tables in the [AP Invoice eForm configuration files](#) section of this document.

1. Create the following work queues in your process:

- APO Invoice Entry
- APO NonPO Invoice GL
- APO PO Invoice Matching
- APO Invoice Upload
- APO Error
- APO NonPO Invoice Approval

2. Create the following system queues in your process:

- APO Validate/Route Invoice
- APO NonPO Invoice Validation

## Configure iScript files and routing actions

You must configure routing rules to use in conjunction with the form in your workflow process. The AP Invoice eForm installer created the custom properties used in routing rule setup.

The procedures in this section are not exhaustive and do not outline every method you can use to set up the routing rules and iScript files specific to your particular AP Invoice Form implementation.

For more information on using Workflow Designer, refer to ImageNow Workflow Designer Help available from the Perceptive Software Customer Support Portal.

## Create an Invoice Entry Validation routing rule

To create an Invoice Entry Validation routing rule in your invoice workflow process, complete the following steps.

1. From the **Workflow Designer** task pane, click **Actions**.
2. In the task pane, click **Manage Actions**.
3. In the **Action Settings** dialog box, on the **Rules** tab, click **New**.
4. In the **Rules Editor** dialog box, complete the following substeps.
  1. In the **Rule name** box, type **Invoice Validation**.
  2. In the **Rule type** list, click **Routing rule**.
  3. Complete the following procedures, detailed in the following sections.
    - [Create a rule to route back for header error conditions](#)
    - [Create a routing rule for non-PO invoices](#)
    - [Create a routing rule for PO-based invoices within tolerance](#)
    - [Create a routing rule for PO-based invoices outside of tolerance](#)

### Create a rule to route back for header error conditions

In the event of an error header message, you can route documents back to the APO Invoice Entry queue. For more information on validation and some possible error messages, refer to the [Configure validation within the eForm](#) section. Complete the following steps.

1. In the **Rules Editor** dialog box, under **Statements**, click **New**.
2. In the **Statement Editor** dialog box, complete the following substeps:
  1. In the **Statement name** box, type **Error Condition – Route Back**.
  2. Under **Conditions**, click the **Add** icon.
  3. In the **Add Condition** dialog box, complete the following substeps:
    1. In the **Constraint by** list, select **Custom property**.
    2. In the **Type** list, select **Normal**.
    3. In the **Field** list, select **AP Header Valid**.
    4. In the **Operator** list, select **starts with**.
    5. In the **Value** box, type **ERR**.

6. Click **OK** to close the **Add Condition** dialog box.
4. Under **Actions**, click **Add** and select **Route to**.
5. In the **Queues** dialog box, complete the following substeps:
  1. In the **Business Process** list, select your workflow process.
  2. Under **Available queues**, select **APO Invoice Entry** or the name of the queue in your process where invoice header data is entered and then click **Add**.
  3. Click **OK** until you return to the **Statement Editor** dialog box.

## Create a routing rule for non-PO invoices

To route non-PO invoices to a specific queue, such as the APO Non-PO Invoice GL queue, create a routing rule. Complete the following steps.

**Note** This statement is necessary only if you are processing both PO-based and non-PO invoices with the form.

1. In the **Rules Editor** dialog box, under **Statements**, click **New**.
2. In the **Statement Editor** dialog box, complete the following substeps:
  1. In the **Statement name** box, type **Non-PO Invoice**.
  2. Under **Conditions**, click the **Add** icon.
  3. In the **Add Condition** dialog box, complete the following substeps:
    1. In the **Constraint by** list, select **Document key**.
    2. In the **Type** list, select **Normal**.
    3. In the **Field** list, select **Field5** or the field in your indexing structure where the PO number is stored.
    4. In the **Operator** list, select **is equal to**.
    5. In the **Value** box, select **Default**.
    6. Click **OK** to close the **Add Condition** dialog box.
  4. Under **Actions**, click **Add** and select **Route to**.
  5. In the **Queues** dialog box, complete the following substeps:
    1. In the **Business Process** list, select your workflow process.
    2. Under **Available queues**, select **APO Non-PO Invoice GL** or the name of the queue in your process where GL coding is entered, and then click **Add**.
    3. Click **OK** until you return to the **Statement Editor** dialog box.

## Create a routing rule for PO-based invoices within tolerance

To route PO-based invoices within tolerance to a specific queue, such as the APO Invoice Upload queue, complete the following steps.

1. In the **Rules Editor** dialog box, under **Statements**, click **New**.
2. In the **Statement Editor** dialog box, complete the following substeps:
  1. In the **Statement name** box, type **PO-Based Invoice – Within Tolerance**.
  2. Under **Conditions**, click the **Add** icon.
  3. In the **Add Condition** dialog box, complete the following substeps:
    1. In the **Constraint by** list, select **custom property**.
    2. In the **Type** list, select **Normal**.
    3. In the **Field** list, select **PO Variance Factor**.
    4. In the **Operator** list, select **is equal to**.
    5. In the **Value** box, type **0**.
    6. Click **OK** to close the **Add Condition** dialog box.
  4. Under **Actions**, click **Add** and select **Route to**.
  5. In the **Queues** dialog box, complete the following substeps:
    1. In the **Business Process** list, select your workflow process.
    2. Under **Available queues**, select **APO Invoice Upload** or the name of the queue in your process for invoice upload and then click **Add**.
    3. Click **OK** until you return to the **Statement Editor** dialog box.

## Create a routing rule for PO-based invoices outside of tolerance

To route PO-based invoices outside of tolerance to a specific queue, such as the APO PO Invoice Matching queue, complete the following steps.

**Note** This statement is necessary only if you are processing PO invoices with the form.

1. In the **Rules Editor** dialog box, under **Statements**, click **New**.
2. In the **Statement Editor** dialog box, complete the following substeps:
  1. In the **Statement name** box, type **PO-Based Invoice – Outside of Tolerance**.
  2. Under **Conditions**, click the **Add** icon.
  3. In the **Add Condition** dialog box, complete the following substeps:
    1. In the **Constraint by** list, select **custom property**.
    2. In the **Type** list, select **Normal**.
    3. In the **Field** list, select **PO Variance Factor**.
    4. In the **Operator** list, select **is not equal to**.
    5. In the **Value** box, type **0**.
    6. Click **OK**.

4. Under **Actions**, click **Add** and select **Route to**.
5. In the **Queues** dialog box, complete the following actions:
  1. In the **Business Process** list, select your workflow process.
  2. Under **Available queues**, select **APO PO Invoice Matching** or the name of the queue in your process for PO invoice matching and then click **Add**.
  3. Click **OK** until you return to the **Workflow Designer** task pane.

## Create a GL Coding Validation routing rule

Complete the following actions to create a GL Coding Validation routing rule.

1. Verify that you are working with the correct workflow process containing the rule you want to modify.
2. From the **Workflow Designer** task pane, click **Actions**.
3. In the task pane, click **Manage Actions**.
4. In the **Action Settings** dialog box, on the **Rules** tab, click **New**.
5. In the **Rules Editor** dialog box, complete the following substeps:
  1. In the **Rules name** box, type **Invoice Validation**.
  2. In the **Rule type** list, select **Routing rule**.
  3. Under **Statements**, complete the following substeps to include a statement for **Check for Errors in GL Coding**:
    1. Click **New**.
    2. In the **Statement Editor** dialog box, complete the following substeps:
      1. In the **Statement name** box, type **Check for Errors in GL Coding**.
      2. Under **Conditions**, click the **Add** icon.
      3. In the **Add Condition** dialog box, complete the following substeps:
        1. In the **Constraint by** list, select **Custom property**.
        2. In the **Type** list, select **Normal**.
        3. In the **Field** list, select **AP Detail Valid**.
        4. In the **Operator** list, select **starts with**.
        5. In the **Value** box, type **ERR**.
        6. Click **OK** to close the **Add Condition** dialog box.
    4. Under **Actions**, click **Add** and select **Route to**.
    5. In the **Queues** dialog box, complete the following substeps:
      1. In the **Business Process** list, select your workflow process.
      2. Under **Available queues**, select **APO NonPO Invoice GL** or the name of the queue in your process where GL coding is entered and then click **Add**.
      3. Click **OK** until you return to the **Workflow Designer** task pane.

## Configure workflow queue properties

After you add the server validation script and create the routing rules used in the invoice form workflow process, you must modify the properties of the workflow queues and add the form to user work queues.

Assign the AP invoice form to the following queues:

- APO Invoice Entry
- APO NonPO Invoice GL
- APO NonPO Invoice Approval
- APO PO Invoice Matching

Apply the AP\_Validate\_Server.js script as an inbound action to the following queues:

- APO Validate/Route Invoice
- APO NonPO Invoice Validation
- APO Invoice Upload

Apply Invoice Entry Validation routing rule as a within queue action to the following queues:

- APO Validate/Route Invoice
- GL Coding Validation

## Configure validation within the eForm

AP Invoice eForm includes both client-side and server-side methods of data validation.

**Client-side validation** allows your users to verify their data in the eForm user interface instantly. The user also receives feedback as he or she types values into fields, or upon clicking the “Validate” button on the form. The user interface error messages and field-flagging enables users to correct errors or omissions before saving or routing the eForm.

**Server-side validation** completes eForm verification with the AP\_Validate\_Server.js script, when run as a queue action. This validation process processes all AP Invoice eForm documents that pass through the queue, regardless of whether a user views or modifies document in the user interface. The validation script creates error codes which you can use when creating routing rules.

To configure validation, review and complete the following tasks, as necessary.

### Enable validation for specific roles and queues

#### Client-side validation for specific roles

Client-side validation takes place within queues attributed to a role, such as Invoice Entry. The first step towards allowing any validation to occur in the user interface is to enable the role or roles. Complete the following steps.

1. Navigate to `[drive:]\\inserver6\\etc\\ap` and open **AP\_Skin.xml** with a text editor.
2. Locate the `<roles>` node.
3. For any queue that requires header-level, client-side validation, in the corresponding `<role>` element, set the `header` attribute to **enabled**. To run detail-level validation, enable the corresponding `detail` attribute. In this example, to enable client-side validation in a queue called APO Invoice Entry, you would enable the Invoice Entry role:

```
<role name="Invoice Entry" header="enabled" detail="enabled">
  <queue invoicetype="nonpo" changetabs="true">APO Invoice Entry</queue>
</role>
```

#### Server-side validation for specific queues

Server-side validation occurs within a queue that includes the AP\_Validate\_Server.js script as an action.

### Configure the duplicate invoice check

The duplicate invoice check validates a new invoice against other invoices in ImageNow to determine if it is a duplicate.

**Client action:** The eForm checks for duplicate invoices after the user provides values for the fields used to determine the uniqueness of an invoice. For example, the fields containing Invoice Number and Vendor ID. It provides the error message “A duplicate invoice exists in the system. To resolve the conflict, review the current invoice and make any necessary changes.”

There are two forms of duplicate invoice check available.

- The original **duplicate invoice check** method validates the Invoice Number and Vendor Number values against all other invoices in the system.

- The **advanced duplicate invoice** check method allows you to specify an ImageNow view, the specific eForm fields to examine, the number of values that indicate a duplicate invoice, and a tolerance level for those values.

If you enable both forms of duplicate invoice check, AP Invoice eForm defaults to the original validation method.

## Duplicate invoice check

To configure the original duplicate invoice check validation, complete the following steps.

1. Navigate to **[drive:]\inserver6\etc\ap** and open **AP\_Config.xml** with a text editor.
2. In the **<dupcheck>** and **<dupcheck\_vs1>** elements, set the **enabled** attribute to **true**.
3. The default fields for the **Invoice Number** and **Vendor ID** fields are **tab** (field2) and **f3** (field3). To specify alternate locations for these values in your specific eForm solution, choose from the following values.

- |                             |                                      |
|-----------------------------|--------------------------------------|
| • <b>[folder]or[field1]</b> | • <b>[f5]or[field5]</b>              |
| • <b>[tab]or[field2]</b>    | • <b>[doctype]</b>                   |
| • <b>[f3]or[field3]</b>     | • <b>[drawer]</b>                    |
| • <b>[f4]or[field4]</b>     | • <b>{any custom property}</b>       |
|                             | For example, <b>{Invoice Number}</b> |

## Advanced duplicate invoice check

To configure the advanced duplicate invoice check validation, complete the following steps.

1. Navigate to **[drive:]\inserver6\etc\ap** and open **AP\_Config.xml** with a text editor.
2. In the **<dupcheck>** set the **enabled** attribute to **true**.
3. In the **<statement>** element, complete the following substeps.
  - To enable the advanced duplicate invoice check, set **enabled** attribute to **true**.
  - To specify a custom view to match against literals, set the **view** attribute to the name of the view. The default view is All Documents.
  - To set the number of values that must match to indicate a duplicate invoice, set the **match** attribute with an integer. The default is 2.
4. Between the **<statement>** element tags, specify which fields you want AP Invoice eForm to check. Specify a number greater than or equal to the number of values you specified within the element, and use a **|** delimiter. Use any of the following fields:

- |                                |                       |
|--------------------------------|-----------------------|
| • <b>AddlAmt&lt;1 to 8&gt;</b> | • <b>LocationCode</b> |
| • <b>CompanyNumHdr</b>         | • <b>MiscCharge</b>   |
| • <b>FreightAmount</b>         | • <b>PONumber</b>     |
| • <b>Hdr&lt;1 to 8&gt;</b>     | • <b>TaxAmount</b>    |
| • <b>InvoiceDate</b>           | • <b>TotalAmount</b>  |

- `InvoiceNumber`
- `VendorID`
- `InvoiceReferenceNumber`

- Optional. To specify a tolerance level with an amount or percent, complete one of the following steps.
  - To set a tolerance amount, following the eForm field name, add `#` and an integer representing the variance amount. For example, to specify a Total Amount field variance of 3 currency units, such as \$3.00, type `TotalAmount # 3`. In this instance, if all other conditions applied, a Total Amount value of \$9.50 would match a Total Amount value of \$12.00, and the eForm would indicate a duplicate invoice. You may specify a tolerance amount with decimals, such as `.01` for one US cent.
  - To set a variance percent for matching an eForm field, include `%` and an integer representing the variance percent. For example, to specify a variance percent of 10%, type `TotalAmount % 10`.

**Example:** The default settings create a duplicate invoice check that examines the VendorID, Invoice Number, and Total Amount fields in the All Documents view. The Total Amount field includes a range of +/- 5. If the user enters matching values in two or more fields, the invoice check indicates a duplicate.

```
<statement view="All Documents" match="2" enabled="true">VendorID | InvoiceNumber |
TotalAmount # 5 </statement>
```

## Configure header-level and line-level required fields

Required-field validation verifies that every field you mark as required contains a value.

**Client action:** The eForm flags any blank, required fields with a red box or red highlighting, and provides the error message "Required fields are missing."

**Server action:** The validation script sends the error key `ERR_REQUIRED_FIELDS` to the AP Header Valid or AP Detail Valid custom property.

Complete the following client-side and server-side configuration tasks.

### Client-side validation

Complete the following steps.

- In `AP_Skin.xml`, locate the `<fields>` node.
- For the first field you want to set as required, locate the corresponding `<header>`, `<poline>`, or `<glline>` element. For example, the element for the Business Unit field is:

```
<header label="lblBusinessUnit" field="cboBusinessUnit">
```

- Set the `required` and `visible` attributes to `"true"`, and set the `disabled` attribute to `"false"`.
- To set additional fields as required, repeat the previous steps.

### Server-side validation

Complete the following steps.

- In `AP_Validate_Server.xml`, in the `<required_fields>` element, set the `enabled` attribute to `"true"`.

2. To enable the PO, non-PO, or credit memo header fields, set the corresponding **enabled** attribute to "true". For example, `<pohdr enabled="true">`.
3. To require a header-level field, such as CompanyNumHdr, in the `<worksheet>` element, set the **enabled** attribute to "true".
4. To enable the PO or GL detail fields, set the corresponding **enabled** attribute to "true". For example, `<podetail enabled="true">`.
5. To require a PO or GL line-level field, under the `<podetail>` or `<gldetail>` node, in the corresponding `<worksheet>` element, set the **enabled** attribute to "true". For example, `<worksheet enabled="true" field="ItemNumber">`.

## Configure GL code validation

In AP Invoice eForm, the general ledger code fields appear under the GL DISTRIBUTIONS section. For example, Account, Sub Account, Unit, and Cost Center. In AP\_Skin.xml, you can configure which of these GL codes to display and how the field displays in the user interface. You can also configure two types of validation.

With GL code validation, the eForm can check user-provided values in GL code fields against available, corresponding values in the GL Codes virtual table. In the client, the eForm can also verify the authorization of the user entering values on a field-by-field basis. You can enable these validation methods together or separately.

**Client action:** The eForm flags any fields with an invalid or restricted value with a red box or red highlighting. It displays the error message "GL codes are invalid."

**Server action:** The validation script sends the error key ERR\_GL\_CODES\_INVALID to the AP Detail Valid custom property.

To enable GL code validation, review and complete the following tasks, as needed.

- [Configure the GL Codes source values.](#)
- Optional. [Configure the GL User and GL Code virtual tables with user constraints.](#)
- [Enable the GL code validation.](#)

## Configure the GL Codes source values

To validate the values of GL codes in the eForm, you must configure the GL Codes virtual table with source numbers. In AP\_Skin.xml each `<glline>` element represents a GL Code field, and includes a **source** attribute. This attribute links to the GLNumber column of the GL Code virtual table, indicating which GL Codes are valid for the field. To enable validation of each GL code you use in AP Invoice eForm, complete the following steps.

- In your **GL Code** virtual table, verify or configure source numbers in the **GLNumber** column for each row.
- For client-side validation: in **AP\_Skin.xml**, for each GL code your eForm uses, in the `<glline>` element, specify the **source** attribute with a source number.
- For server-side validation: in **AP\_Validate\_Server.xml**, under `<glcode_validation>`, for each GL code, specify the **source** attribute with the same source number.

For example, referring to the GLNumber column in the example [GL Code](#) table, note that the source number 1 corresponds to the GLCode column values of 1101 and 1102. In this instance, if you set the source attribute for **lblGL1** to 1, then the eForm validates that the user types either 1101 or 1102 into the first GL Code in the eForm, such as **Account**.

AP\_Skin.xml

```
<glline label="lblGL1" showdesc="true" searchable="true" freeform="false" source="1"
line="1" typeaheadchars="2" visible="true" disabled="false" required="true"
width="50px" captionkey="LBL_GLLINE_GL1" descriptionkey="LBL_GLLINE_GL1_DESC" />
```

AP\_Validate\_Server.xml

```
<worksheet enabled="false" field="GL1" source="1"></worksheet>
```

## Configure the GL User and GL Code virtual tables with constraints

Optional. To validate the authorization of a user to provide a specific value in a GL Code, you must configure the GL User and GL Code virtual tables with constraints. This validation occurs only in the client, and does not have a server-side counterpart. Complete the following steps.

1. In your **GL User** virtual table, create a row for every AP Invoice eForm user. Provide the username in the **UserName** column.
2. Specify a pair of constraints for each user, for example **Department\_01** and **KCMO**.

**Note** Valid constraints include strings or numbers.

3. In your **GL Code** virtual table, to configure which users can access each GL code value, specify the constraint columns for each row.

**Note** To allow all users to access a GL code, use the constraint **ALL** for both columns.

For example, referring to the Constraint columns in the [GL Code](#) and [GL User](#) tables, note that user JDOE could use the values 1101 and 1102, while user JJONES could use the value 1103. Each code has a different source, which indicates that users JDOE and JJONES would use the values in different GL Code fields, such as Account vs. Unit.

## Enable the GL code validation

To enable the value and (optional) user constraint validation of your GL codes, complete the following steps.

### Client-side validation

- In **AP\_Skin.xml**, under the **<fields>** node, for each GL code you want to validate in the client, set the **freeform** and **disabled** attributes to **false**, and set **visible** to **true**. For example,

```
<glline label="lblGL1" field="txtGL1" showdesc="true" searchable="true"
freeform="false" source="1" line="1" typeaheadchars="2" visible="true"
disabled="false" required="true" width="50px" captionkey="LBL_GLLINE_GL1"
descriptionkey="LBL_GLLINE_GL1_DESC" />
```

**Note** The GL code elements include the **label** attribute **lblGL<1 to 24>**.

### Server-side validation

For each GL Code you enabled for validation in the previous section, enable validation in **AP\_Validate\_Server.xml**. Complete the following steps.

1. In **AP\_Validate\_Server.xml**, in the `<glcode_validation>` element, set the `enabled` attribute to `"true"`.
2. Under the `<glcode_validation>` element, for each GL code you enabled in **AP\_Skin.xml**, enable the corresponding `<worksheet>` element.

## Configure GL account validation

GL account validation verifies the data your users enter in the eForm against the corresponding column of the GL Account virtual table. For more information on this virtual table, refer to the [GL Account](#) section.

**Client action:** The eform flags the GL line and displays the error message “One or more invalid GL Accounts.”

**Server action:** The validation script sends the error key ERR\_GLACCT to the AP Detail Valid custom property.

### Client-side validation

For each field on which you want to enable validation, complete the following step.

- In **AP\_Skin.xml**, under the `<fields>` node, for each GL account you want to validate, set the `visible` and `required` attributes to `"true"` and set `disabled` to `"false"`. For example,

```
<glline label="lblGLAcct" field="txtGLAcct" line="1" visible="true" maxlength="10"
required="true" disabled="false" width="50px" captionkey="LBL_GLACCT_GLACCT" />
```

**Note** The GL account elements include the seven elements between the `label="lblGLBusUnit"` and `label="lblGLAcct6"`.

### Server-side validation

1. In **AP\_Validate\_Server.xml**, enable the `<glaccount_validation>` element.
2. Under the `<glaccount_validation>` node, for each GL account you want to validate, enable the corresponding `<worksheet>` element. For example,

```
<worksheet enabled="true" field="GLAcct"></worksheet>
```

## Configure the Invoice Total field validation

AP Invoice eForm allows you to validate the Invoice Total field in PO, non-PO, and credit memo invoices.

**PO** invoice validation compares the header-level Invoice Total against the line-level Invoice Lines amount total.

**Non-PO** and **credit memo** validation methods compare the header-level Invoice Total against the GL Distributions amount total.

### Configure amount validation and distribution in PO invoices

For PO invoices, AP Invoice eForm validates the header-level Invoice Total amount against the line-level Invoice Lines amount total. To configure this validation, you can choose whether to include Sales Tax, Freight, Misc, and eight other available “additional charges” in the Invoices Lines amount total.

In other words, if you include additional amounts, the eForm verifies:

$$(\text{Invoice Total}) = (\text{Invoice Lines}) + (\text{included additional charges})$$

You can also allow users to distribute additional amounts from the header-level fields into general ledger-level fields, under GL DISTRIBUTIONS.

**Client action:** The eForm flags the **To Be Distributed** amounts with red boxes. It displays the error message "Invoice Amount not fully distributed."

**Server action:** The validation script sends the error key ERR\_DISTR to the AP Detail Valid custom property.

To include or distribute additional charge fields, complete the following steps, as needed.

### Client-side validation

- In **AP\_Skin.xml**, under the **<po>** node, under the **<distribution total>** node, set the desired additional charge element to **true**. The default for every node is false.

For example, to have AP Invoice eForm include the Sales Tax and Freight values in the Invoice Lines amount validated against Invoice Total, configure the following elements:

```
<po>
  <polines>true</polines>
  <change_remitto>true</change_remitto>
  <distributiontotal>
    <include_freight>true</include_freight>
    <include_tax>true</include_tax>
    <include_misc>false</include_misc>
    <include_addl_amt1>false</include_addl_amt1>
    <include_addl_amt2>false</include_addl_amt2>
    <include_addl_amt3>false</include_addl_amt3>
    <include_addl_amt4>false</include_addl_amt4>
    <include_addl_amt5>false</include_addl_amt5>
    <include_addl_amt6>false</include_addl_amt6>
    <include_addl_amt7>false</include_addl_amt7>
    <include_addl_amt8>false</include_addl_amt8>
  </distributiontotal>
</po>
```

### Client-side distribution

- In **AP\_Skin.xml**, under the **<fields>** node, in the **<header>** element for the additional charges you want to distribute, set the **distributable** attributes to **"true"**.

For example, to distribute the Tax Amount field, but not the Freight Amount field, configure the elements as shown in this example.

```
<header label="lblTaxAmount" field="txtTaxAmount" visible="true" align="right"
maxlength="13" distributable="true" captionkey="LBL_SALES_TAX"/>
<header label="lblFreightAmount" field="txtFreightAmount" visible="true"
align="right" maxlength="13" distributable="false" captionkey="LBL_FREIGHT"/>
```

## Server-side validation

1. In **AP\_Validate\_Server.xml**, enable the **<po\_distribution\_validation>** node.
2. Under the **<po\_distribution\_validation>** node, for each additional amount you included in **AP\_Skin.xml**, enable the corresponding **<include\_>** element. For example,

```
<po_distribution_validation enabled="true">
  <include_tax enabled="true" distributed="true"/>
  <include_freight enabled="true" distributed="true"/>
  <include_misc enabled="false" distributed="false"/>
  <include_addl_amt1 enabled="false" distributed="false"/>
  <include_addl_amt2 enabled="false" distributed="false"/>
  <include_addl_amt3 enabled="false" distributed="false"/>
  <include_addl_amt4 enabled="false" distributed="false"/>
  <include_addl_amt5 enabled="false" distributed="false"/>
  <include_addl_amt6 enabled="false" distributed="false"/>
  <include_addl_amt7 enabled="false" distributed="false"/>
  <include_addl_amt8 enabled="false" distributed="false"/>
</po_distribution_validation>
```

## Configure Invoice Total validation in non-PO invoices or credit memos

For non-PO invoices and credit memos, AP Invoice eForm validates the header-level Invoice Total amount against the line-level GL Distributions amount total. To customize this validation, you can choose whether to include Sales Tax, Freight, Misc, and the other eight available additional charges in the GL Distributions amount total.

In other words, if you include additional amounts, the eForm verifies that:

$$(\text{Invoice Total}) = (\text{GL Distributions}) + (\text{included additional charges})$$

**Client action:** The eForm flags the **To Be Distributed** amounts with red boxes. It displays the error message “Invoice Amount not fully distributed” or “Distribution percentages do not total to 100%.”

**Server action:** The validation script sends the error key **ERR\_DISTR** or **ERR\_PCT\_DISTR** to the AP Detail Valid custom property.

To include an additional charges, complete the following steps.

## Client-side validation

- In **AP\_Skin.xml**, under the **<nonpo>** or **<credit>** node, under the **<distribution total>** node, set the desired additional charge element to **true**. The default for every node is false.

For example, to have AP Invoice eForm include the Sales Tax and Freight values in the GL Distributions amount validated against the Invoice Total amount, configure the following elements.

```
<nonpo>
  <distributiontotal>
    <include_freight>true</include_freight>
    <include_tax>true</include_tax>
    <include_misc>>false</include_misc>
    <include_addl_amt1>>false</include_addl_amt1>
    <include_addl_amt2>>false</include_addl_amt2>
    <include_addl_amt3>>false</include_addl_amt3>
    <include_addl_amt4>>false</include_addl_amt4>
    <include_addl_amt5>>false</include_addl_amt5>
    <include_addl_amt6>>false</include_addl_amt6>
    <include_addl_amt7>>false</include_addl_amt7>
    <include_addl_amt8>>false</include_addl_amt8>
```

```

        </distributiontotal>
    </nonpo>

```

## Server-side validation

1. In **AP\_Validate\_Server.xml**, enable the **<nonpo\_distribution\_validation>** or **<credit\_distribution\_validation>** node, as needed.
2. Under the **<nonpo\_distribution\_validation>** or credit node, for each additional amount you included in **AP\_Skin.xml**, enable the corresponding **<include\_>** element. For example,

```

<nonpo_distribution_validation enabled="true">
    <include_tax enabled="true"/>
    <include_freight enabled="true"/>
    <include_misc enabled="false"/>
    <include_addl_amt1 enabled="false"/>
    <include_addl_amt2 enabled="false"/>
    <include_addl_amt3 enabled="false"/>
    <include_addl_amt4 enabled="false"/>
    <include_addl_amt5 enabled="false"/>
    <include_addl_amt6 enabled="false"/>
    <include_addl_amt7 enabled="false"/>
    <include_addl_amt8 enabled="false"/>
</nonpo_distribution_validation>

```

## PO/invoice currency validation

If you use the eForm with Intelligent Capture data extraction, the PO/invoice currency validation verifies the PO currency matches the invoice currency. This validation is not configurable. If the PO virtual table includes values in the currency column, then the validation runs. The currency list is disabled in invoice documents. Users should correct a mismatch on the PO level.

**Client action:** The eForm displays the error message “The purchase order currency does not match the invoice currency.”

**Server action:** The validation script sends the error key **ERR\_CURR\_MATCH** to the AP Header Valid custom property.

## VAT validation

For more information on enabling VAT validation, refer to the [Enable VAT amount validation](#) section.

## Configure value added tax (VAT) within the eForm

Incorporating VAT into your AP Invoice eForm solution allows your users to process and validate invoices and credit memos that include VAT amounts, codes and jurisdictions. To use VAT with AP Invoice eForm, review and complete the following procedures.

### Create and enable the VAT codes

For both purchase order and non-purchase order invoices, AP Invoice eForm provides fields for assigning VAT codes for each line item. The VAT code consists of two components: the jurisdiction and the VAT rate code. To configure AP Invoice eForm to accept line-level VAT code assignment, complete the following steps.

#### Create or update the virtual tables

To create VAT-specific tables and update your existing virtual tables to include the basic VAT-specific columns, complete the following steps.

1. Create the **VAT Code** virtual table with **Jurisdiction**, **VATCode** and **VATRate** columns. For more information, refer to the [VAT Code](#) section.
2. Update the **Business Unit** virtual table with a **Jurisdiction** column. For more information, refer to the [Business Unit](#) section.
3. Optional. Update the **Purchase Order Lines** virtual table with **Jurisdiction** and **VATCode** columns. For more information, refer to the [Purchase Order Lines](#) section.

#### Update AP\_VirtualTable\_Updater.xml

To enable the VAT-specific columns you created in the previous section, complete the following step.

- In **AP\_VirtualTable\_Updater.xml**, for each **<recordstructure>** and **<column>** element your solution requires, set the **enabled** attribute to **"true"**. The following example illustrates enabling the VAT Code table and its columns.

```
<recordstructure recordtype="VAT_CODE" table="Z_APW_APVAT_Code" enabled="true">
  <column customprop="Z_APW Active" enabled="true"/>
  <column customprop="Z_APW VAT Jurisdiction" enabled="true"/>
  <column customprop="Z_APW VAT Code" enabled="true"/>
  <column customprop="Z_APW VAT Rate" enabled="true"/>
  <column customprop="Z_APW VAT Description" enabled="true"/>
</recordstructure>
```

#### Configure AP\_Skin.xml

In the **AP\_Skin.xml** file, you can enable the global use of VAT in the eForm, at the line level.

1. Navigate to **[drive:]\\inserver6\\etc\\ap** and open **AP\_Skin.xml** with a text editor.
2. To display the VAT code at the line level, for each of the following elements, set the **visible** attribute to **"true"**.
  - **<poline label="IbIPOLVATCode" field="cboPOLVATCode"/>**
  - **<glline label="IbGLVATCode" field="cboGLVATCode"/>**

3. Optional. To allow users to view the Jurisdiction code at the line level, in the following elements, set the `visible` attribute to `"true"`.
  - `<poline label="IbIPOLVATJurisdiction" field="txtPOLVATJurisdiction"/>`
  - `<glline label="IbIGLVATJurisdiction" field="txtGLVATJurisdiction"/>`
4. Optional. To allow users to view the VAT amount at the line level, in the following elements, set the `visible` attribute to `"true"`.
  - `<poline label="IbIPOLVATAmount" field="txtPOLVATAmount"/>`
  - `<glline label="IbIGLVATAmount" field="txtGLVATAmount"/>`
5. Save and close **AP\_Skin.xml**.

## Enable the VAT amount summary

For each distinct VAT code referenced on an invoice, AP Invoice eForm can summarize the total VAT amounts. In the eForm user interface, the VAT Summary section provides a simple, compiled total of the VAT Base and VAT Amount for each VAT Code. To enable the VAT amount summary logic, and optionally display the VAT Summary section in the eForm, complete the following steps.

**Note** To enable VAT sum validation, with or without displaying the VAT amount summary in the eForm, refer to the following [Enable VAT amount validation](#) section.

1. Navigate to `[drive:]\inserver6\etc\ap` and open **AP\_Skin.xml** with a text editor.
2. To enable VAT summation logic in the eForm, under `<vatsummary>`, set `<enable>` to `true`.
3. Optional. To display the VAT Summary section in AP Invoice eForm, set `<showsection>` to `true`.
4. Save and close **AP\_Skin.xml**.

## Enable VAT amount validation

To enable validation of the line-level VAT Amount sums against the header-level VAT Total amount, complete the following procedures.

### Update the Currency Codes virtual table

Due to rounding rate percentages, the sum total of the line-level VAT Amounts and the header-level VAT Total often contain a degree of variation. To specify an acceptable level of variation for your solution, update the Currency Codes virtual table. For more information, refer to the [Currency Codes](#) section.

Complete the following step.

- Update the **Currency Codes** virtual table with **VATVarianceAmount** or **VATVariancePercent** columns.

**Note** If neither column exists, AP Invoice eForm validation expects an exact match between the line-level VAT Amounts and the header-level VAT Total. If you configure both columns, AP Invoice eForm uses the **VATVariancePercent** column.

### Update AP\_VirtualTable\_Updater.xml

To enable the **VATVarianceAmount** and **VATVariancePercent** columns in the Currency Codes table, complete the following step.

- In **AP\_VirtualTable\_Updater.xml**, in the `<recordstructure recordtype="Z_APW_APCurrency">` element, set the `enabled` attribute to `"true"` for the following columns.

```
<column customprop="Z_APW VAT Variance Amount" enabled="true"/>
<column customprop="Z_APW VAT Variance Percent" enabled="true"/>
```

## Configure AP\_Skin.xml

To enable VAT summation and validation in the eForm user interface, configure **AP\_Skin.xml**. Complete the following steps.

1. Navigate to `[drive:]\inserver6\etc\ap` and open **AP\_Skin.xml** with a text editor.
2. To convert the Tax Amount label to a VAT Total label in the eForm header, in the `<header label="lblTaxAmount">` element, set the `captionkey` attribute to `LBL_VAT_TOTAL`.
3. To enable VAT summation in the eForm, under `<vatsummary>`, set `<enable>` to `true`.
4. To enable VAT validation in the eForm, under `<vatsummary>`, set `<validatesummary>` to `true`.
5. Save and close **AP\_Skin.xml**.

## Configure AP\_Validate\_Server.xml

With server-side VAT validation, you can compare the header-level VAT Total field against the line-level VAT Amount fields in a workflow validation queue. Complete the following steps.

**Note** Server-side validation allows you to validate VAT amounts for each document processed, without requiring the user to complete the validation step in the eForm user interface.

1. Navigate to `[drive:]\inserver6\etc\ap` and open **AP\_Validate\_Server.xml** with a text editor.
2. In the `<vatsummary_validation>` element, set the `enabled` attribute to `"true"`.
3. Save and close **AP\_Validate\_Server.xml**.
4. Optional. In **Workflow Designer**, create a routing rule to route back for VAT validation-specific errors, using the error code `ERR_VAT_TOTAL_MATCH` for the **Value**. For more information on routing rules, refer to the [Create a rule to route back for header error conditions](#) section.

## Associate the vendor VAT registration numbers

To associate vendor VAT registration numbers (VAT IDs) with vendors and enable the Vendor VAT ID field, complete the following procedures.

### Create the VAT Registration Number (VAT ID) Vendor virtual table

To associate one or more VAT registration numbers (VAT IDs) per vendor and enable an optional default, complete the following steps.

1. Create the **VAT\_ID\_VENDOR** virtual table with the required **BusUnit**, **Vendor**, and **VAT ID** columns. For more information, refer to the [VAT Registration Number \(VAT ID\) Vendor](#) section.
2. Optional. To create a default VAT ID value for each vendor, update the **VENDOR** virtual table with a **VAT ID** column. For more information, refer to the [Vendor](#) section.

**Note** The user can modify the default value in the eForm by manually keying a new value, or by selecting a value from the VAT Registration Number virtual table using the search feature.

## Configure AP\_Skin.xml

To enable the Vendor VAT ID header-level field, configure AP\_Skin.xml. Complete the following steps.

1. Navigate to **[drive:]\inserver6\etc\ap** and open **AP\_Skin.xml** with a text editor.
2. For the following element, set the **visible** attribute to **true**.

- `<header label="lblVendorVATID" field="txtVendorVATID"/>`

3. Save and close **AP\_Skin.xml**.

## Associate the sold-to VAT Registration Numbers

To associate sold-to VAT registration numbers (VAT IDs) with business units and enable the Sold To VAT ID field, complete the following procedures.

### Create the VAT Registration Number (VAT ID) Business Unit virtual table

To create VAT-specific tables and update your existing virtual tables to include VAT-required columns, complete the following steps.

1. Create the **VAT\_ID\_BUS\_UNIT** virtual table with the required **BusUnit**, and **VAT ID** columns. For more information, refer to the [VAT Registration Number \(VAT ID\) Business Unit](#) section.
2. Optional. To create a default VAT ID value for each business unit, update the **BUS\_UNIT** virtual table with a **VAT ID** column. For more information, refer to the [Business Unit](#) section.

**Note** The user can modify the default value in the eForm by manually keying a new value, or by selecting a value from the VAT Registration Number virtual table using the search feature.

## Configure AP\_Skin.xml

In the AP\_Skin.xml file, you can enable the global use of VAT in your eForm, at the header level.

1. Navigate to **[drive:]\inserver6\etc\ap** and open **AP\_Skin.xml** with a text editor.
2. For the following element, set the **visible** attribute to **true**.

- `<header label="lblSoldToVATID" field="txtSoldToVATID"/>`

3. Save and close **AP\_Skin.xml**.

## Enable the VAT Compliance check box

You can customize the AP Invoice eForm to include a check box that signifies that the invoice follows all rules and regulations around the handling of invoices with VAT. To enable this feature, complete the following steps.

1. Navigate to **[drive:]\inserver6\etc\ap** and open **AP\_Skin.xml** with a text editor.
2. For the following element, set the **visible** attribute to **"true"**.

- `<header label="lblVATCompliant" field="chkVATCompliant"/>`

**Note** You can set the **required** attribute to **true**, however the user must still select the check box in the eForm.

3. Save and close **AP\_Skin.xml**.

## Configure add on cost (AOC) within the eForm

The add on cost (AOC) feature allows you to configure the way you categorize and distribute Sales Tax, Freight, Misc. Charge, and other additional amount charges. These charges appear on PO invoices, but are not included in the purchase order. For Lawson users, this configuration also allows you to export AOC codes with distributed amounts, including GL Code or GL Account information, to the MA540 file.

For example, you could assign a default AOC code and accompanying description to your header-level “Misc. Charge” field, such as “HA – Handling.” During PO invoice processing, your user keys an amount into the “Misc. Charge” field. The GL Distributions section automatically populates the default AOC code, description, distribution amount, and any other default GL information you prefer, such as Account number.

At that point, your user could retain the default AOC code for the Misc. Charge value, or select an alternate code from the AOC list. The user can insert and modify additional distribution lines under the AOC code. For example, the user could split a handling charge across multiple departments, cost centers, or accounting units. The AOC feature supports the ability to incorporate both GL Code and GL Account fields in your GL Distributions section.

To configure AOC codes and PO GL distributed-line exports for Lawson, refer to the following sections.

### Create and enable the AOC codes

AP Invoice eForm provides the ability to assign an AOC code for each PO GL header-level field amount. In the eForm, the AOC list populates a concatenated AOC code and optional description. The list can import these values from the AOC GL Code table, the AOC GL Account table, or both.

To configure AP Invoice eForm with AOC functionality, complete the following tasks.

#### Create the AOC virtual tables

To use AOC, create an AOC GL Code table, an AOC GL Account table, or both, depending on your current usage of the GL Code and GL Account virtual tables. If you use both AOC tables, the AOC list under GL Distributions contains each unique AOC Code and its description. If you use both GL Code and GL Account fields, use matching sets of AOC codes and optional descriptions.

The eForm completes field-level validation with the GL Code and GL Account tables. To pass validation, every code or account field used in the AOC tables must also exist in the corresponding GL Code or GL Account table.

Complete the following steps, as necessary for your solution.

- Create the **AOC GL Account** virtual table with **Table Name**, **Active**, **AOC Code**, and any optional columns. For more information, refer to the [AOC GL Account](#) section.
- Create the **AOC GL Code** virtual table with **Table Name**, **Active**, and **AOC Code**, and any optional columns. For more information, refer to the [AOC GL Code](#) section.

#### Update AP\_VirtualTable\_Updater.xml

To enable the import of the AOC-specific data you created in the previous section, complete the following step.

- In **AP\_VirtualTable\_Updater.xml**, for each `<recordstructure>` and `<column>` element your solution requires, set the `enabled` attribute to `“true”`. The following example illustrates how to enable the AOC GL Account virtual table and select columns.

```
<recordstructure recordtype="AOC_GL_ACCT" table="Z_APW_APAOCGLAccount"
enabled="true">
  <column customprop="Z_APW Active" enabled="true"/>
  <column customprop="Z_APW AOC Code" enabled="true"/>
  <column customprop="Z_APW AOC Description" enabled="true"/>
  <column customprop="Z_APW Business Unit ID" enabled="true"/>
  <column customprop="Z_APW GL Acctg Unit" enabled="true"/>
  <column customprop="Z_APW GL Account" enabled="true"/>
  <column customprop="Z_APW GL Sub Account" enabled="true"/>
  <column customprop="Z_APW GL Account Description" enabled="true"/>
  <column customprop="Z_APW GL Account 5" enabled="false"/>
  <column customprop="Z_APW GL Account 6" enabled="false"/>
  <column customprop="Z_APW GL Account 7" enabled="false"/>
  <column customprop="Z_APW GL Account 8" enabled="false"/>
</recordstructure>
```

## Enable GL distribution with AOC codes in AP\_Skin.xml

In the AP\_Skin.xml file, enable the distribution of the header-level additional charges and assign an optional default AOC code to each. Complete the following steps.

1. Navigate to **[drive:]\\inserver6\\etc\\ap** and open **AP\_Skin.xml** with a text editor.
2. To enable the AOC list for the GL Distributable fields in PO invoices, under **<fields>**, in the **<addl entries label="lblAOCCode">** element, set the **visible** attribute to **"true"**.
3. To configure distribution, complete the following steps.
  1. Under **<fields>**, in the **<header>** element, such as **<header label=lblFreightAmount>**, set the **distributable** attribute to **"true"**.
  2. Optional. Set the **aoc** attribute with a default AOC code from either the AOC GL Code or AOC GL Account virtual table. For example:

```
<header label="lblFreightAmount" field="txtFreightAmount" visible="true"
align="right" maxlength="13" distributable="true" captionkey="LBL_FREIGHT"
aoc="FR"/>
```

### Notes

- If the GL Distributions section is visible, the optional columns in the AOC GL Code and AOC GL Account tables, such as business unit ID, automatically populate the fields.
  - Users have the ability to select any AOC code from the list while processing the PO invoice. Select another AOC code to update the values populated in each GL Code or GL Account field. The eForm completes validation using the GL Code and GL Account tables.
4. Save and close **AP\_Skin.xml**.
  5. To commit the **AP\_Skin.xml** updates, run the **AP\_Maintenance.js** script.

## Enable multi-line GL distribution exports for Lawson

Lawson customers can export each line-item amount in the GL Distributions section as a separate line to the MA540 file. To enable the export of distributed amounts, complete the following tasks.

### Enable PO GL distributions in AP\_Config.xml

To enable the export of PO GL distributed lines to Lawson as individual, line-item amounts, configure the AP\_Config.xml file. Complete the following steps.

1. Navigate to `[drive:]\inserver6\etc\ap` and open **AP\_Config.xml** with a text editor.
2. Under **<AP\_Invoice\_Export>**, set **<Misc\_Line>** to **true**.

**Note** For all nodes under **<ExportSections>**, the **true/false** settings are case-sensitive.

3. Save and close **AP\_Config.xml**.

### Configure the miscellaneous line export in AP\_Export.xml

Configure your Lawson version of the AP\_Export.xml file to export distributed amounts and AOC codes. With this functionality, each PO GL distributed amount exports on its own line for the MA540 file. You configure the values that accompany the distributed amount. For example, company number, vendor ID, invoice number, AOC code, and associated GL codes.

#### Notes

- The order of the **<field>** elements determines the column order in the CSV file.
- The **sequence** attribute is for your reference only: it does not control the column order.
- You may export both header-level amounts, with the **<misc>** node, and distributed amounts, with the **<misc\_line>** node.

Complete the following steps.

1. Navigate to `[drive:]\inserver6\etc\ap` and open **AP\_Export.xml** with a text editor.
2. Under **<misc\_line>**, to configure a **<field>** element for every column you require in your CSV file, complete any of the following steps, as needed.
  - To configure a blank column, in the corresponding **<field>** element, do not assign a value to the **worksheet** or **constant** attributes. Verify the **name** attribute is unique within the **<misc\_line>** node.
  - To create a column with a constant value, in a **<field>** element, create or modify the attribute **constant="variable"**. For example, you may configure the MA-REC-TYPE column to carry the value **M**, to denote a Lawson M-line.
 

```
<field name="MA-REC-TYPE" sequence="1" type="alpha" length="1" constant="M"/>
```
  - To create a column that contains the **Workflow Item Creation Date**, set the **constant** attribute to **"{{scandate}}"**, and set the **length** attribute to **"10"**. The date exports in MM/DD/YYYY format.
  - To create a column with incrementing integers, set the **constant** attribute to **"{{sequence}}"**, and set the **length** attribute to a minimum of **"2"**, such as for integers 1 to 99. The sequence iterates for each exported line in the **<misc\_line>** node, and resets for each PO Invoice exported concurrently.

- To create a column that maps to an eForm value, set the `worksheet` attribute to the eForm field. For a list of available worksheet attributes, refer to the following AP\_Export.xml Worksheet Attributes table in [Appendix C](#).

```
<field name="MA-VENDOR" sequence="3" type="alpha" length="9"
worksheet="VendorID"/>
```

- Save and close **AP\_Export.xml**.

## Globalization

ImageNow makes it possible to run ImageNow anywhere a supported language is spoken, allowing users to interact with the product in a format and display they understand.

### Supported languages, countries and currencies

AP Invoice eForm supports the following languages, country specific languages and currencies:

Supported Language	Language Code	Supported County	Country Code	Supported Currency	Currency Code
Dutch	nl	Belgium Netherlands	BE NL	Euro Euro	EUR
English  English (en) is the default language.	en	Australia Canada Great Britain India Ireland South Africa United States	AU CA GB IN IE ZA US	Australian dollar Canadian dollar British pound Indian rupee Euro South African rand US dollar	AUD CAD GBP INR EUR ZAR USD
French	fr	Belgium France Switzerland	BE FR CH	Euro Euro Swiss frank	EUR CHF
German	de	Germany Switzerland	DE CH	Euro Swiss frank	EUR CHF
Italian	it	Italy Switzerland	IT CH	Euro Swiss frank	EUR CHF
Brazilian Portuguese	pt	Brazil	BR	Brazilian reel	BRL

Supported Language	Language Code	Supported County	Country Code	Supported Currency	Currency Code
Spanish	es	Argentina	AR	Argentinian peso	ARS
		Columbia	CO	Columbian peso	COP
		Ecuador	EC	Ecuadorian sucre	ECS
		Mexico	MX	Mexican peso	MXP
		Spain	ES	Euro	EUR

**Note** Additional country locales may be supported. Contact your Perceptive Software representative for more information.

## Additional supported currencies

In addition, AP Invoice eForm supports the following currencies:

Country	Supported Currency	Currency Code
Denmark	Danish Krone	DKK
Finland	Euro	EUR
Sweden	Swedish krona	SEK

## Configuration considerations

**Note** For specific instructions on configuring your language, refer to the *AP Invoice eForm Installation and Setup Guide*.

When you configure ImageNow in a global environment, configure the following settings:

- Windows regional settings
- Keyboard configuration
- Language Bar

For specific instructions on changing these settings, refer to Windows Help and Support.

### Windows regional settings

To display ImageNow data in your ImageNow Client or WebNow browser according to your locale, configure Windows regional settings to the language and locale you are running, such as Spanish in Spain or Spanish in Mexico. The formatting guidelines for each region vary. Refer to Microsoft Windows help for more information.

### Keyboard configuration

You can easily define a list of languages to be available for your own use, by adding or removing input languages in the Regional and Language Options in the Microsoft Windows Control Panel. For example, two users in the same department can configure a different input language on their respective devices. One user can use an English (United States) keyboard layout and a German (Germany) keyboard layout, while the other user can use English (United Kingdom) keyboard layout and a German (Swiss German) keyboard layout. This allows each user to insert or display text in both languages.

### Language Bar

After you customize your keyboard layout, use the Microsoft Windows Language Bar located on your desktop or Taskbar to switch between the input languages you have added. You can move the Language Bar anywhere on your desktop or minimize it to the Taskbar.

When you change your language of choice using the Language Bar, the selected language applies only to the currently active software program. For example, if your default language is English (United States), you open Microsoft Word, and you then change your language to German (Germany), the setting only applies to Microsoft Word. All other programs you subsequently open use English (United States). Always verify that you are using the correct keyboard layout and language of choice when switching between programs.

## Appendix A: Add a custom currency example

The following steps show you how to add support for an additional currency to the AP Invoice eForm. In the example below, the AP Invoice eForm includes the Venezuelan Bolívar (VEB).

1. Add a record to the **VENDOR** virtual table with Venezuela's country code, **VE**.
2. Add a record to the **CURRENCY** virtual table for the **Venezuelan Bolívar**.

	A	B	C	D	E	F	
1	<b>TableName</b>	<b>BusUnitID</b>	<b>CurrencyCode</b>	<b>CurrencyName</b>	<b>PayGroup</b>	<b>Active</b>	
2	CURRENCY	1234	VEB	Venezuelan Bolívar	7000	A	

3. Add a record to the **COUNTRYCURRENCY** virtual table, tying Venezuela's country code, **VE**, to the Venezuelan Bolívar currency, **VEB**.

	A	C	B	D	E	
1	<b>TableName</b>	<b>CurrencyCode</b>	<b>CountryCode</b>	<b>LocaleCode</b>	<b>Active</b>	
2	COUNTRYCURRENCY	VEB	VE	es-VE	Y	

4. In a text editor, open the **AP\_Custom\_Cultures.xml** file, and complete the following substeps:
  1. Add a new culture section, with formatting details, for the currency you just added.
  2. In the new culture section, specify the group separator and the decimal separator.

```
<cultures>
  <culture name="es-VE" base="es" language="es">
    <numberFormat groupSeparator="." decimalSeparator=",">
    </numberFormat>
  </culture>
</cultures>
```

**Note** The culture name in the XML file must match the LOCALECODE in the COUNTRYCURRENCY virtual table.

3. Save and close the **AP\_Custom\_Cultures.xml** file.
5. Start **INTool** by accessing a command prompt and changing directories to **\inserver6\bin**.
- Note** To use the 64-bit version of INTool, change the directory to **\inserver6\bin64**.
6. Enter the following command to run the **AP\_Maintenance.js** script and commit the changes.

```
intool --cmd run-iscript --file AP_Maintenance.js
```

7. Complete the following substeps to verify the currency formatting:
  1. Open the AP Invoice eForm.
  2. Select the vendor you added in the previous steps and enter a value in the **Invoice Amount** field.
  3. Verify the value follows the correct currency formatting for the Venezuelan Bolívar.

## Appendix B: User interface labels, tooltips, and messages

### User interface labels

The following table lists the available captionkeys and user interface captions for customization in the AP\_Custom\_Cultures.xml file.

In the case of eForm fields, such as Vendor Name, the table also lists the corresponding worksheet attribute for the AP\_Export.xml file. Some fields, such as Business Unit, can export more than one value, such as the business unit number (CompanyNumHdr) and the business unit name (CompanyNameHdr). You can map one or both fields in the AP\_Export.xml file independently.

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_LOADING	Loading form	
LBL_BUSINESS_INFO	Business Information	
LBL_INV_INFO	Invoice Information	
LBL_GL_DIST	GL Distributions	
LBL_INV_LINES	Invoice Lines	
LBL_BUS_UNIT	Business Unit	CompanyNumHdr CompanyNameHdr
LBL_LOC	Location	LocationCode LocationName
LBL_PO_NUM	PO Number	PONumber
LBL_PO_AMT	PO Amount	POAmount POCurrency
LBL_VEND_NAME	Vendor Name	VendorName
LBL_VEND_ID	Vendor Number	VendorID
LBL_REMIT_TO	Remit To	RemitTo
LBL_VEND_ADDR	Vendor Number	VendorAddress
LBL_VEND_LOC	Vendor Location	VendorLocation VendorLocationName
LBL_VEND_SHORT_NAME	Short Name	VendorShortName

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_VEND_PAY_TERMS	Payment Terms	VendorPaymentTerms
LBL_VENDOR_VAT	VAT	VendorVAT
LBL_VEND_FIELD_1	Field 1	VendorField1
LBL_VEND_FIELD_2	Field 2	VendorField2
LBL_INV_NUM	Invoice Number	InvoiceNumber
LBL_INV_DATE	Invoice Date	InvoiceDate
LBL_INV_DUE_DATE	Invoice Due Date	InvoiceDueDate
LBL_INV_AMT	Invoice Amount	TotalAmount
LBL_INV_REF_NUM	Invoice Ref Num	InvoiceReferenceNumber
LBL_CREDIT_REASON	Credit Reason	CreditMemoReason
LBL_SALES_TAX	Sales Tax <b>Note</b> If utilizing VAT with your eForm, use LBL_VAT_TOTAL rather than this caption key.	TaxAmount
LBL_VAT_TOTAL	Value added tax (VAT) <b>Note</b> If utilizing VAT with your eForm solution, use this caption key rather than LBL_SALES_TAX.	TaxAmount
LBL_FREIGHT	Freight	FreightAmount
LBL_MISC_CHARGE	Misc. Charge	MiscCharge
LBL_PAY_IMMEDIATELY	Pay Immediately	PayImmediately
LBL_ADDL_AMT_1	Addl Amt 1	AddlAmt1
LBL_ADDL_AMT_2	Addl Amt 2	AddlAmt2
LBL_ADDL_AMT_3	Addl Amt 3	AddlAmt3
LBL_ADDL_AMT_4	Addl Amt 4	AddlAmt4
LBL_ADDL_AMT_5	Addl Amt 5	AddlAmt5

Captionkey attribute in AP_Skin.xml  Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_ADDL_AMT_6	Addl Amt 6	AddlAmt6
LBL_ADDL_AMT_7	Addl Amt 7	AddlAmt7
LBL_ADDL_AMT_8	Addl Amt 8	AddlAmt8
LBL_SUT_CODE	SUT Code	SUTCode
LBL_SUT_APPLY	SUT Apply	SUTApplyCode SUTApplyCodeName
LBL_UUT_CODE	UUT Code	UUTCode
LBL_TAX_EXEMPT	Tax Exempt	TaxExempt
LBL_CURRENCY	Currency	Currency CurrencyName
LBL_PAY_GROUP	Pay Group	PayGroup
LBL_SPECIAL_HANDLING	Special Handling	SpecialHandling
LBL_PAYMENT_COMMENTS	Payment Comments	Comments
LBL_HDR_1	Hdr1	Hdr1
LBL_HDR_2	Hdr2	Hdr2
LBL_HDR_3	Hdr3	Hdr3
LBL_HDR_4	Hdr4	Hdr4
LBL_HDR_5	Hdr5	Hdr5
LBL_HDR_6	Hdr6	Hdr6
LBL_HDR_7	Hdr7	Hdr7
LBL_HDR_8	Hdr8	Hdr8
LBL_HDR_9	Hdr9	Hdr9
LBL_HDR_10	Hdr10	Hdr10
LBL_POLINE_MATCH_SELECT_ALL	Select All	POLineMatchSelectAll

Captionkey attribute in AP_Skin.xml  Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_POLINE_MATCH	Match	UpdateHost
LBL_POLINE_TAXABLE_SELECT_ ALL	Select All	POLineTaxableSelectAll
LBL_POLINE_TAXABLE	Taxable	POLineTaxable
LBL_POLINE_NUM	Line #	LineNumber
LBL_POLINE_ITEM_NUM	Item #	ItemNumber
LBL_POLINE_DESCRIPTION	Description	ItemDescription
LBL_POLINE_QUANTITY	Quantity	Quantity
LBL_POLINE_UOM	UOM	UOM
LBL_POLINE_UNITPRICE	Unit Price	UnitPrice
LBL_POLINE_EXT_AMT	Extended Amount	ExtendedAmount
LBL_POLINE_POLINE_NUM	PO Line #	PO_LineNumber
LBL_POLINE_PO_QUANTITY	PO Quantity	PO_Quantity
LBL_POLINE_PO_UOM	PO UOM	UOM
LBL_POLINE_PO_UNIT_PRICE	PO Unit Price	UnitPrice
LBL_POLINE_PO_1	PO1	PO1
LBL_POLINE_PO_2	PO2	PO2
LBL_POLINE_PO_3	PO3	PO3
LBL_POLINE_PO_4	PO4	PO4
LBL_POLINE_PO_5	PO5	PO5
LBL_POLINE_PO_6	PO6	PO6
LBL_POLINE_PO_7	PO7	PO7
LBL_POLINE_PO_18	PO8	PO8
LBL_POLINE_PO_9	PO9	PO9

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_POLINE_PO_1 0	PO10	PO10
LBL_GLLINE_TAXABLE_SELECT_A LL	Select All	GLLineTaxableSelectAll
LBL_GLLINE_TAXABLE	Taxable	
LBL_GLLINE_PCT	Pct	LinePct
LBL_GLLINE_AMT	Amount	LineAmount
LBL_GLLINE_GL1	Account	GL1
LBL_GLLINE_GL1_DESC	Account Description	GL1Desc
LBL_GLLINE_GL2	Unit	GL2
LBL_GLLINE_GL2_DESC	Unit Description	GL2Desc
LBL_GLLINE_GL3	Sub Unit	GL3
LBL_GLLINE_GL3_DESC	Sub Unit Description	GL3Desc
LBL_GLLINE_GL4	Sub Acct	GL4
LBL_GLLINE_GL4_DESC	Sub Acct Description	GL4Desc
LBL_GLLINE_GL5	Activity	GL5
LBL_GLLINE_GL5_DESC	Activity Description	GL5Desc
LBL_GLLINE_GL6	GL6	GL6
LBL_GLLINE_GL6_DESC	Activity Description	GL6Desc
LBL_GLLINE_GL7	GL7	GL7
LBL_GLLINE_GL7_DESC	Activity Description	GL7Desc
LBL_GLLINE_GL8	GL8	GL8
LBL_GLLINE_GL8_DESC	Activity Description	GL8Desc
LBL_GLLINE_GL9	GL9	GL9
LBL_GLLINE_GL9_DESC	Activity Description	GL9Desc

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_GLLINE_GL10	GL10	GL10
LBL_GLLINE_GL10_DESC	Activity Description	GL10Desc
LBL_GLLINE_GL11	GL11	GL11
LBL_GLLINE_GL11_DESC	Activity Description	GL11Desc
LBL_GLLINE_GL12	GL12	GL12
LBL_GLLINE_GL12_DESC	Activity Description	GLDesc
LBL_GLLINE_GL13	GL13	GL13
LBL_GLLINE_GL13_DESC	Activity Description	GL13Desc
LBL_GLLINE_GL14	GL14	GL14
LBL_GLLINE_GL14_DESC	Activity Description	GL14Desc
LBL_GLLINE_GL15	GL15	GL15
LBL_GLLINE_GL15_DESC	Activity Description	GL15Desc
LBL_GLLINE_GL16	GL16	GL16
LBL_GLLINE_GL16_DESC	Activity Description	GL16Desc
LBL_GLLINE_GL17	GL17	GL17
LBL_GLLINE_GL17_DESC	Activity Description	GL17Desc
LBL_GLLINE_GL18	GL18	GL18
LBL_GLLINE_GL18_DESC	Activity Description	GL18Desc
LBL_GLLINE_GL19	GL19	GL19
LBL_GLLINE_GL19_DESC	Activity Description	GL19Desc
LBL_GLLINE_GL20	GL20	GL20
LBL_GLLINE_GL20_DESC	Activity Description	GL20Desc
LBL_GLLINE_GL21	GL21	GL21

Captionkey attribute in AP_Skin.xml  Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_GLLINE_GL21_DESC	Activity Description	GL21Desc
LBL_GLLINE_GL22	GL22	GL22
LBL_GLLINE_GL22_DESC	Activity Description	GL22Desc
LBL_GLLINE_GL23	GL23	GL23
LBL_GLLINE_GL23_DESC	Activity Description	GL23Desc
LBL_GLLINE_GL24	GL24	GL24
LBL_GLLINE_GL24_DESC	Activity Description	GL24Desc
LBL_GLACCT_BUS_UNIT	Business Unit	CompanyNum
LBL_GLACCT_ACCTG_UNIT	Acctg Unit	GLUnit
LBL_GLACCT_GLACCT	GL Acct	GLAcct
LBL_GLACCT_SUB_ACCT	Sub Account	SubAcct
LBL_GLACCT_ACCT_DESC	Account Description	GLAcctDesc
LBL_GLACCT_5	GL Acct 5	GLAcct5
LBL_GLACCT_6	GL Acct 6	GLAcct6
LBL_TOTAL_DIST	Total Distributed	
LBL_TO_BE_DIST	To Be Distributed	
LBL_VALIDATE	Validate	
LBL_PO_DATE	PO Date	
LBL_PO_DATE_MATCHED	Date Matched	
LBL_PO_DATE_CLOSED	Date Closed	
LBL_SEARCH_TITLE	Search	
LBL_POLINES_TITLE	PO Lines	
LBL_CREDIT_MEMO	Credit Memo #	InvoiceNumber

Captionkey attribute in AP_Skin.xml  Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_CREDIT_DATE	Credit Date	InvoiceDate
LBL_CREDIT_AMT	Credit Amount	CreditAmount
LBL_VEND_ID	Vendor Number	VendorID
LBL_VEND_NAME	Vendor Name	VendorName
LBL_SEARCH_VENDOR_ADDR	Address	
LBL_REMIT_TO	Remit To	RemitTo
LBL_MIN_CHAR	(minimum 3 characters)	
LBL_OK	OK	
LBL_CANCEL	Cancel	
LBL_LINES_ADD	Add a line	
LBL_LINES_DELETE	Delete this line	
LBL_INVTYPE_PO	PO	
LBL_INVTYPE_NON_PO	Non-PO	
LBL_INVTYPE_CREDIT	Credit	
LBL_ADDL_ENTRY_DESC	Description	AOCCode
LBL_ADDL_ENTRY_AMT	Extended amount	AOCDescription
LBL_POLINE_VAT_JURISDICTION	Jurisdiction	PO_VATJurisdiction
LBL_POLINE_VAT_CODE	VAT Code	POLVATRate POLVATCode
LBL_POLINE_VAT_AMOUNT	VAT Amount	POLVATAmount
LBL_POLINE_PO_VAT_JURISDICTION	Jurisdiction	POLVATJurisdiction
LBL_POLINE_PO_VAT_CODE	VAT Code	PO_VATCode PO_VATRate
LBL_GLLINE_VAT_JURISDICTION	Jurisdiction	GLVATJurisdiction

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption	Worksheet attribute in AP_Export.xml
LBL_GLLINE_VAT_CODE	VAT Code	GLVATCode GLVATRate
LBL_GLLINE_VAT_AMOUNT	VAT Amount	GLVATAmount
LBL_SEARCH_VENDOR_VAT	VAT	
LBL_SEARCH_VENDOR_FIELD1	Field 1	
LBL_SEARCH_VENDOR_FIELD2	Field 2	
LBL_VAT_SUMMARY	VAT SUMMARY	
LBL_VAT_SUMMARY_CODE	VAT Code	VATSummaryCode
LBL_VAT_SUMMARY_BASE	VAT Base	VATSummaryBase
LBL_VAT_SUMMARY_AMOUNT	VAT Amount	VATSummaryAmount
LBL_VAT_SUMMARY_TOTAL	VAT Sum	VATSummaryTotal
LBL_VAT_VEND_ID	VAT Vendor ID	
LBL_SOLD_TO_VAT_ID	Sold-To VAT ID	SoldToVATID
LBL_VAT_COMPLIANT	VAT Compliant	VATCompliant
LBL_TOTALS	TOTAL DISTRIBUTION	
LBL_VAT_TOTAL	VAT Total	

## Tooltips

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption
TOOLTIP_DATE_FORMAT	Please enter date in {datemask1} or {datemask2} format. <b>Note</b> AP Invoice eForm replaces {datemask1} and {datemask2} with the appropriate date format, as determined by AP_Custom_Cultures.xml. For example, "Please enter date in MMDDYYYY or MMDDYY format."
TOOLTIP_CLOSE	Close

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface caption
TOOLTIP_IMG_PO_LINES_DIALOG	View PO Lines
TOOLTIP_IMG_GL_ACCT_SEARCH_WINDOW	Look Up Account
TOOLTIP_IMG_PO_SEARCH_WINDOW	Look Up PO Number
TOOLTIP_IMG_VENDOR_SEARCH_WINDOW	Look Up Vendor
TOOLTIP_IMG_VENDOR_REMIT_TO_WINDOW	Look Up Remote To
TOOLTIP_IMG_GL_SEARCH_WIND	Look Up Account
TOOLTIP_IMG_VENDOR_VAT_ID_SEARCH_WINDOW	Look Up Vendor VAT ID
TOOLTIP_IMG_SOLD_TO_VAT_ID_SEARCH_WINDOW	Look Up Sold-To VAT ID

## Error and confirmation messages

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface captions
ERR_GEN	Error running eForm. Please check the log files for more information.
ERR_NONE	No errors.
ERR_DISTR	Invoice Amount not fully distributed.
ERR_PCT_DISTR	Distribution percentages do not total to 100%.
ERR_GLACCT	One or more invalid GL Accounts.
ERR_GLACCT_MISSING	Enter GL Distribution Entries.
ERR_EXTAMT	Line Item amounts do not total Extended Amounts. Please check distribution.
ERR_PREVIOUSLY_EXPORTED	This invoice was previously processed.
ERR_REQUIRED_FIELDS	Required fields are missing.
ERR_GL_CODES_INVALID	GL codes are invalid.

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface captions
ERR_SEARCH_LIMIT	incomplete
ERR_NO_MATCH	No match found.
ERR_VENDOR_NOT_FOUND	Vendor not found.
ERR_DUP_CHECK	The duplicate invoice detection process requires configuration. Contact your system administrator.
ERR_DUP_INVOICE	A duplicate invoice exists in the system. To resolve the conflict, review the current invoice and make any necessary changes.
ERR_PO_CLOSED	This PO is closed.
ERR_PO_NUM_NOT_FOUND	PO number not found.
ERR_CURR_MATCH	The purchase order currency does not match the invoice currency.
ERR_CONFIG	Configuration error: {value1} permission = '{value2}'. Changing to '{value3}'.
ERR_PO_GL_DISTR	GL distributions not fully distributed
ERR_DATE_FORMAT	Enter a date in {datemask1} or {datemask2} format. <b>Note</b> AP Invoice eForm replaces {datemask1} and {datemask2} with the appropriate date format, as determined by AP_Custom_Cultures.xml. For example, "Please enter date in MMDDYYYY or MMDDYY format."
ERR_CURRENCY_FORMAT	Enter an amount in ###{decimalmask1}## format. <b>Note</b> AP Invoice eForm replaces ###{decimalmask1}## with the appropriate decimal format, as determined by AP_Custom_Cultures.xml.
ERR_DECIMAL_FORMAT	Enter a decimal in ###{decimalmask1}## format. <b>Note</b> AP Invoice eForm replaces ###{decimalmask1}## with the appropriate decimal format, as determined by AP_Custom_Cultures.xml.
ERR_PERCENT_FORMAT	Enter a percentage in ###{decimalmask1}## format. <b>Note</b> AP Invoice eForm replaces ###{decimalmask1}## with the appropriate decimal format, as determined by AP_Custom_Cultures.xml.
ERR_VAT_TOTAL_MATCH	The header-level Value Added Tax does not match the line-level Value Added Tax.
ERR_CURR_MATCH	The purchase order currency does not match the invoice currency.

Captionkey attribute in AP_Skin.xml Key attribute in AP_Custom_Cultures.xml	Default English user interface captions
ERR_SOLD_TO_VAT_ID_NOT_FOUND	No VAT ID found for the selected business unit.\n\n Enter a VAT ID or contact a system administrator to request that a VAT ID be added for this business unit. <b>Note</b> \n\n places two new lines between the first and second sentences.
ERR_VENDOR_VAT_ID_NOT_FOUND	No VAT ID found for the selected vendor.\n\n Enter a VAT ID or contact a system administrator to request that a VAT ID be added for this vendor. <b>Note</b> \n\n places two new lines between the first and second sentences.
CONFIRM_PO_NUMBER_CHANGE	The form will replace all existing lines related to the PO number. Do you want to change the PO number?
CONFIRM_PO_NUMBER_CHANGE_POLINEMATCHING	The form will clear all captured invoice data. Do you want to change the PO number?
MESSAGE_PO_NUMBER_CLEAR	To clear all PO and GL lines, click the Clear Form Data button above.
TEXT_CRITERIA_VENDOR_SEARCH	Enter vendor name.
TEXT_CRITERIA_GL_ACCT_SEARCH	Enter account description.
TEXT_CRITERIA_PO_SEARCH	Enter vendor ID.
TEXT_MATCH	match
TEXT_MATCHES	matches

## Index

add on cost	
AOC GL Account table.....	100
AOC GL Code table.....	100
codes.....	100
overview .....	100
AP workflow process .....	81
AP_CustomValidation_Client.js.....	76
AP_GetData_Custom.js .....	77
AP_Validate_Server.js .....	72
AP_VirtualTable_Updater.xml .....	21
attribute	
caption.....	38
captionkey .....	39, 107
key .....	107
worksheet .....	57, 107
body onload event .....	76
body onunload event .....	77
Brainware .....	<i>See</i> Intelligent Capture
CSV files .....	80
custom logic.....	29
custom properties.....	23
database lookups.....	76
date format.....	40, 55
datemask.....	40, 115, 117
dbcall.....	77
dbcall method .....	77
dupcheck .....	41
Duplicate invoices.....	41
error queue.....	72
incremental update .....	22
inputpath variable .....	30
Intelligent Capture.....	43
INTool.....	22, 55, 80, 106
invoice header data.....	36
iScript.....	82
JavaScript.....	76
JavaScript Event Handlers .....	76
Lawson	
MA540 file .....	102
Match Audit Exceptions .....	37
multi-line GL distribution .....	102
PO matching .....	43
Purge routine .....	28
reserved characters .....	74
routing actions .....	82
user exits.....	76
user roles .....	36
validate.....	28
validation.....	76
validation script.....	72
Value added tax.....	96, 100
amount summary .....	97
amount validation.....	97
codes.....	96
Compliance check box .....	99
ID98, 99	
registration numbers, sold-to.....	99
registration numbers, vendor .....	98
Variance factor.....	72
VAT .....	<i>See</i> Value added tax
Virtual Table CSV Files .....	7
Virtual Table Updater .....	21
virtual tables.....	21, 22, 71
VSL query .....	41
workflow queue roles.....	73
WorksheetStylesheet.css .....	29
XML reserved characters.....	74

