

ImageNow AP Invoice eForm

Supplemental Guide

Version: 10.3.0.x

Compatible with ImageNow Versions: 6.5.1.149 to 6.7.x

The logo for perceptive software, featuring the word "perceptive" in a bold, lowercase sans-serif font, followed by "software" in a lighter, lowercase sans-serif font. A green arrow points to the right, positioned above the text.

Written by: Product Documentation, R&D
Date: January 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

- Virtual Tables..... 6**
 - Required files..... 6
 - Business Unit*..... 6
 - GL Account* 7
 - GL Code*..... 8
 - Location* 9
 - Purchase Orders*..... 9
 - Purchase Order Lines*..... 10
 - Vendor* 12
 - Optional files..... 13
 - Country Currency*..... 13
 - Currency Codes* 14
 - GL User*..... 15
 - Payment Terms*..... 15
 - Special Handling*..... 16
 - SUT Codes* 16
 - SUT Apply Codes* 17
 - Vendor Location*..... 17
 - Virtual Table Updater configuration file 18
 - AP_VirtualTable_Updater.xml* 18
 - Virtual tables updates 19
 - Archive processed files..... 19
- Document types and custom properties 20**
 - Invoice 20
 - Credit Memo 21
- AP Invoice eForm file directories 22**
 - \inserver6\etc\AP 22
 - \inserver6\temp\AP 23
 - \inserver6\script 24
 - \inserver6\form\[build folder]..... 26
 - \inserver6\form\data_definition 27
- AP Invoice eForm log files 27**
- AP Invoice eForm configuration files 28**

AP_Skin.xml	28
AP_Config.xml	34
AP_Export.xml	46
AP_Validate_Server.xml	51
AP_Custom_Cultures.xml	53
AP_VirtualTable_Updater.xml	55
Configure workflow for AP Invoice eForm	56
About the reindex and custom mapping script	56
About the validation script	56
About workflow queue roles	57
Reserved characters in XML and forms	58
Custom Logic	59
Custom functions	59
Global variables	59
Add custom business logic to the form	60
JavaScript event handlers	60
Call a server-side iScript	61
Enhance the validation iScript	62
Enhance the export iScript	63
<i>Field level sample</i>	63
<i>Global level sample</i>	63
Demonstration Setup for AP Invoice eForm	64
Load Data	64
Create AP workflow process	65
Configure iScript files and routing actions	66
Create an Invoice Entry Validation routing rule	66
<i>Error Condition during Route Back</i>	66
<i>Non-PO Invoice</i>	67
<i>PO-based Invoice – Within Tolerance</i>	68
<i>PO-based Invoice – Outside of Tolerance</i>	68
Create a GL Coding Validation routing rule	69
Configure workflow queue properties	70
Globalization	71
Supported languages, countries and currencies	71
Additional supported currencies	72

Configuration considerations 72

Windows regional settings 72

Keyboard configuration 72

Language Bar 72

Appendix A: Add a custom currency example 73

Appendix B: Form fields and string values 74

Index 83

Virtual Tables

You need several Comma Separated Value (CSV) files to load the virtual tables. Each file has a different format. Do not change the TableName field for any of these files.

Required files

Business Unit

The BUS_UNIT table contains an entry for every Business Unit associated with AP invoices. The following columns are required:

- TableName
- Active
- BusUnitID
- BusUnitName
- VendorGroup

Most business applications have a Vendor Group value. If none is available, use the letter A as the value.

	A	B	C	D	E	
1	TableName	Active	BusUnitID	BusUnitName	VendorGroup	
2	BUS_UNIT	Y	1234	ACME Construction	ACME	
3	BUS_UNIT	Y	4500	CGI Corporation	CGI	
4	BUS_UNIT	Y	9001	International Retail	2	
5	BUS_UNIT	Y	4321	LGE Corporation	4321	
6	BUS_UNIT	Y	9000	UK Retail Ltd	1	

GL Account

This table contains an entry for every GL Account that is referenced within the AP Invoice process for Non-PO Invoices.

	A	B	C	D	E	F	G	H	I
1	TableName	Active	BusUnitID	ACCTGUNI	GLAcct	GLSubAcct	GLAcctDesc	GLACCT5	GLACCT6
2	GL_ACCT	Y	1234	101	17900	100	Office Supplies - Misc	3370	7071
3	GL_ACCT	Y	1234	101	17900	200	Office Supplies - Copiers	3370	7071
4	GL_ACCT	Y	1234	101	17900	300	Office Supplies - Printers	3370	7071
5	GL_ACCT	Y	1234	101	22500	100	Maintenance – Buildings	3375	7072
6	GL_ACCT	Y	1234	101	22500	200	Maintenance – Grounds	3375	7072
7	GL_ACCT	Y	1234	101	22500	100	Maintenance – HVAC	3375	7072

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 by setting the default the value to a null value. GLAcct5 and GLAcct6 (columns H and I) are optional elements that provide additional flexibility. If you do not use these elements, default the value to 0 (zero).

Note If you are using the GL Account table but not the GL Account description field, you must still enter a value for this element. The validation process indicates an invalid description entry if the GLAcctDesc (column G) value is null.

GL Code

The GLCODE table includes an entry for every general ledger code. The following columns are required:

- TableName
- Constraint1
- Constraint2
- GLNumber
- GLCode
- Active

Each combination of Constraint1, Constraint2, GLNumber and GLCode must be unique. The GL User Table defines which GL Codes a user can access.

You must define GLNumber and Description for GLNumber 1. Every entry must have a GLNumber, however, the GLNumber does not need to be unique for each entry.

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

The Description column is optional.

	A	B	C	D	E	F	G	
1	TableName	Constraint1	Constraint2	GLNumber	GLCode	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	1	1103	Livestock	Y	
5	GLCODE	Department_21	KCMO	1	1104	Laptops	Y	

Location

The LOCATION table contains an entry for every Location associated with AP invoices. The following columns are required:

- TableName
- SetID
- LocCode
- LocName
- Active

The following columns are optional:

- Sales-Use-Tax-Code
- Ultimate-Use-Tax-Code

	A	B	C	D	D	E	G	
1	TableName	SetID	LocCode	LocName	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	

A default value is not required for either of these fields. 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.

Purchase Orders

Note This file is required if using eForm for PO-based invoices.

The PO table contains an entry for every open Purchase Order. The following columns are required:

- TableName
- BusUnitID
- PONumber
- VendorID
- POAmount
- DateClosed
- RemitTo

Use the Date Closed to control what has already been paid. RemitTo is the Vendor Remit To Address Sequence Number.

	A	B	C	D	E	F	G	H	
1	TableName	BusUnitID	PONumber	VendorID	POAmount	PODate	DateClosed	RemitTo	
2	PO	1234	650000109	3004	127.3	7/2/2008		2	
3	PO	1234	650000110	3004	158.17	7/8/2009		1	
4	PO	1234	650000201	3004	584.17	8/1/2009		1	
5	PO	1234	650001001	1001	7300.8	7/20/2009		1	
6	PO	1234	650001012	1001	538.8	8/1/2009		1	
	PO	1234	650000920	2000	1150	8/2/2007		1	

Purchase Order Lines

Note This file is required if using eForm for PO-based invoices.

The PO_LINE table contains a line item entry for every Purchase Order loaded into the PO Table. The following columns are required:

- TableName
- BusUnitId
- PONumber
- POLineNumber
- POItemNumber
- POItemDesc
- POItemQty
- POItemUOM
- POItemUnitPrice
- POItemExtAmt

QTY received less the QTY Invoiced (or matched). This calculation needs to be verified with the customer based on their purchase order system and the data being stored within the PO line items. POItemQty might vary slightly based on the type of matching being performed. The POItemQty column reflects one of the following states, depending on the matching type:

- 2-way is quantity ordered
- 3-way is the quantity received but not invoiced
- 4-way is the 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, this may need to be calculated using business application data.

The DateClosed and Taxable columns are optional.

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

	A	B	C	D	E	F
1	TableName	BusUnitID	PONumber	POLineNumber	POItemNumber	POItemDesc
2	PO_LINE	1234	650000109	1	B100	Binder Clips
3	PO_LINE	1234	650000110	2	CPW 16X	Copier Paper 200 Sheets
4	PO_LINE	1234	650000201	1	AAG-Desk	At-A-Glace Desk Pack
5	PO_LINE	1234	650001001	2	PPCAC	Plastic Paper Clips
6	PO_LINE	1234	650001012	3	TTD114	Things to Do Pads
7	PO_LINE	1234	650000920	1	1-PN	10 Penny Nails

	G	H	I	J	K	L
1	POItemQty	POItemUOM	POItemUnitPrice	POItemExtAmt	Taxable	DateClosed
2	20	BOX	2.49	49.8	Y	
3	5	BOX	15.5	77.5	Y	
4	10	BOX	6.39	63.9	N	
5	20	BOX	3.19	63.8	N	
6	5	BOX	3.89	19.45	N	
7	10	BOX	1.88	18.8		

Vendor

The VENDOR table contains an entry for every active Vendor. The following columns are required:

- TableName
- Active
- Vendor_Group
- VendorID
- VendorName
- Remit_To

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.

The following columns are optional:

- Address1
- Address2
- City
- State
- PostalCode
- CountryCode
- VendorShortName
- PaymentTerms
- VAT
- Field1
- Field2

	A	B	C	D	E	F	G	H
1	TableName	Active	Vendor Group	VendorID	VendorName	Address1	Address2	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	ACME	1001	JOHNSON LUMBER	2356 MAIN		SEATTLE

	I	J	K	L	M	N	O	P	Q
1	State	PostalCode	RemitTo	CountryCode	VendorShort Name	Payment Terms	VAT	Field1	Field2
2	IL	60600	1	US	GENSUP	Net 30			
3	NY	86888	2	US	GENSUP	Net 30			
4	LA	23842	1	US	JLUMBER	Net 30			

Optional files

Country Currency

The COUNTRYCURRENCY table supports the mapping of a country to a currency. This table is optional, however, if the table is not populated, the form follows the number and percentage formatting rules of the language in which the form is originally presented.

The following columns are required:

- TableName
- CountryCode
- CurrencyCode
- LocaleCode

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 code.

	A	C	B	D	E	
1	TableName	CurrencyCode	CountryCode	LocaleCode	Active	
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	UK	en-GB	Y	
12	COUNTRYCURRENCY	INR	IN	en-IN	Y	

Currency Codes

The CURRENCY table contains an entry for all the Currency Codes by Business Unit. The following columns are required:

- TableName
- BusUnitID
- CurrencyCode
- CurrencyName
- Active

The PayGroup column is optional.

	A	B	C	D	E	F	
1	TableName	BusUnitID	CurrencyCode	CurrencyName	PayGroup	Active	
2	CURRENCY	1234	USD	US Dollar	7000	A	
3	CURRENCY	1234	GBP	Pound	8100	A	
4	CURRENCY	1234	EUR	Euro	7000	A	
5	CURRENCY	1234	MXN	Peso	7000	A	
6	CURRENCY	4500	USD	US Dollar	8100	A	

GL User

The GLUSER table includes an entry for every general ledger user. The following columns are required:

- TableName
- UserName
- Active

Each GL User has access to the GL Codes based on the constraints defined in Value1 and Value2, which match GL Codes defined in the GL Code table.

The following columns are optional:

- Value1
- Value2

Values you define for Value1 and Value2 must exist in the Constraint1 and Constraint2 columns in the GL Code table. If Value1 and Value2 columns are left blank, the user can only access GL Codes, which have Constraint1 and Constraint2 values of ALL.

Use the following table only if you want to filter GL Codes by User. Otherwise you do not need to use the GL User Table.

	A	B	C	D	E	
1	TableName	UserName	Value1	Value2	Active	
2	GLUSER	JDOE	Department_01	KCMO	Y	
3	GLUSER	JJONES	Department_05	KCMO	Y	

Payment Terms

The PAYMENTTERMS table calculates the voucher payment due date based on the vendor's payment terms criteria and calculation method. The following columns are required:

- TableName
- PaymentTermCode
- PaymentCalcMethod
- PaymentDayOfMonth
- Active

The payment due date is calculated based on the formula designated in PaymentCalcMethod, which can be one of the following options:

- **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	TableName	PaymentTermCode	PaymentCalcMethod	PaymentDayOfMonth	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	

Special Handling

The SPEC_HANDLING table contains an entry for every Special Handling Code that is to be referenced within the AP invoice form process. The following columns are required:

- TableName
- Active
- SpecHandlingCode
- SpecialHandlingDesc

	A	B	C	D	
1	TableName	Active	SpecHandlingCode	SpecialHandlingDesc	
2	SPEC_HANDLING	Y	EXP	Expedite Payment	
3	SPEC_HANDLING	Y	HOLD	Hold for Supplier Pickup	
4	SPEC_HANDLING	Y	ATT	Attach Supporting Documentation	

SUT Codes

The SUT_CODE table contains an entry for all the Sales Use Tax Codes. The following columns are required:

- TableName
- Active
- TaxCode

	A	B	C	
1	Tablename	Active	TaxCode	
2	SUT_CODE	Y	10100	
3	SUT_CODE	Y	11200	

SUT Apply Codes

The SUT_APPLY table contains an entry for all the Sales Use Tax Apply Codes. The following columns are required:

- TableName
- Active
- ApplyCode

	A	B	C	
1	TableName	Active	ApplyCode	
2	SUT_APPLY	Y	YES	
3	SUT_APPLY	Y	NO	

Vendor Location

The VENDOR_LOCATION table contains a location value that relates to the vendor. It is predominately used for PeopleSoft implementations. The following columns are required:

- TableName
- VendorID
- VendorLoc_Code
- VendorLoc_Name
- Active

	A	B	C	D	E	
1	TableName	VendorID	VendorLoc_Code	VendorLoc_Name	Active	
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	

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 solution doesn't require, in the column element, set enabled to "false."

If you are working 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

```
<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 Date" format="YYYYMMDD" enabled="true"/>
  <column customprop="Z_APW PO Date Closed" enabled="true"/>
  <column customprop="Z_APW Vendor Remit To" enabled="true"/>
  <column customprop="Z_APW Vendor Country Code" enabled="true"/>
  <column customprop="Z_APW Vendor Short Name" enabled="true"/>
  <column customprop="Z_APW Vendor Payment Terms" enabled="true"/>
  <column customprop="Z_APW Vendor VAT" enabled="true"/>
  <column customprop="Z_APW Vendor Field 1" enabled="true"/>
  <column customprop="Z_APW Vendor Field 2" enabled="true"/>
</recordstructure>
```

When you run the Virtual Table Updater, it detects the 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 the ImageNow Server is not on the C:\ drive, you use a non-Windows platform, or have an active-active server environment, then 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 perform 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 is considered 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 Virtual Table Updater script (AP_VirtualTable_Updater.js) to archive processed files automatically by moving them to a specified directory. This script does not rename the file, we recommend that you remove the archived CSV files or name the source CSV files with a unique identifier, such as a date and time stamp. If you do not include a unique identifier, the system will process the CSV file, but it will not move it to the archive directory.

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 date that the invoice is issued.
Invoice Due Date	Date	The date that the invoice is due.
Invoice Amount	Number	The total amount of the invoice.
Invoice Tax Amount	Number	The total amount of taxes of the invoice.
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 status of the invoice export.

Credit Memo

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.
Credit Memo Number	String	The unique number associated with the credit memo.
Credit Date	Date	The date the credit memo is created.
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 status of the invoice export.
Invoice Reference Number	String	The invoice number that is referred to by the credit memo.
Credit Reason	String	The description of why the credit memo is issued.

AP Invoice eForm file directories

The following table details the directories of all of the form files that are installed 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 greatplains. The selected business application is listed 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 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. Refer to the "AP_Skin.xml" 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	This is the default path where CSV data files are stored after they are processed by the AP_VirtualTable_Updater.js iScript.	AP_VirtualTable_Updater.xml	archivepath
C:\inserver6\temp\ap\error	This is the default path where CSV data files are sent if errors are encountered while running the AP_VirtualTable_Updater.js iScript.	AP_VirtualTable_Updater.xml	errorpath
C:\inserver6\temp\ap\export	ImageNow places the exported data files for your business application in this default path when you run the AP_Invoice_Export.js iScript.	AP_Export.xml	outputdir
C:\inserver6\temp\ap\import	Place your CSV data files in this default path to import them to the virtual tables by the AP_VirtualTable_Updater.js iScript.	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 loading the form, the drop-down values that are gathered from the virtual tables by the AP_GetCompanies.js script are cached. When this script runs (the script is called each time the form loads), the values returned to the eForm are saved as a temporary file in \inserver6\temp\AP_GetCompanies.cache.</p> <p>Each time the script runs (the next time the form loads by any user), the logic in AP_GetCompanies.js script decides whether to use cached values from this AP_GetCompanies.cache file or to get fresh values from the virtual tables.</p> <p>This script has the following configurable parameter that determines when the cached values become stale and need to be refreshed.</p> <pre>#define CACHE_DURATION_MINUTES 60</pre> <p>By default, the values are cached for 60 minutes. You can adjust this time as necessary. For example, if the virtual tables are updated each night, then you can set the cache duration to 1440 minutes.</p> <p>Important If you are testing AP_VirtualTable_Updater.js and do not see the changes (new values) reflected in the AP form, delete the cached file from \inserver\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 perform 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. The following files are located 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 a form 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 form installation.

AP_Skin.xml

This file is located in the \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 form.</p> <p>The default is blank.</p>
<customerlogo>logo object name, for example, psi.jpg</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 form.</p> <p>To use a graphic file, you must add the graphic file to the form presentation in ImageNow. This image file must exist in the build folder within the inserver6\worksheets directory. Alternatively, you can use a value for a logo or graphic that is located on the Internet or an intranet. Note that when you use a URL for this value, you can use a URL to a website or to a file.</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"> • read/write access to invoice header information • read/write access to GL coding lines (non-PO invoices) • read/write access to PO lines (PO-based invoices)
<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>
<po>	

Element	Description
<polines>true</polines>	A value for this element is required. Set the value to true if you are configuring for a PO-related form and you want to show and modify PO lines on the form. Otherwise, set the value to false.
<gllines>false</gllines>	A value for this element is required. This function exists because backwards-compatibility is not available at this time. This attribute should be left as false. .
<change_remitto>true</ change_remitto>	This element allows you to specify whether the user is permitted to change the vendor remit to for PO-based invoices.
<distributiontotal>	Each element within the <distributiontotal> element is required. 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. For non-PO invoices, the default value is false.
<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. For non-PO invoices, the default value is false.
<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. For non-PO invoices, the default value is false.
<include_addl_amt1>false</include_addl_amt1>	This element specifies whether to include additional charges in the To Be Distributed calculation. The default value is false. Note You can use the following format to configure up to eight additional charges fields: <include_addl_amt1 >false</include_addl_amt1> <include_addl_amt2 >false</include_addl_amt2> . . . <include_addl_amt8 >false</include_addl_amt8>
</distributiontotal>	
</po>	

Element	Description
<polines>	
<addbutton>>false</addbutton>	This required element is set to false by default. Set this to true if you want to give users the ability to add PO lines. This is not typical.
<delbutton>>false</delbutton>	This required element is set to false by default. Set this to true if you want to give users the ability to delete PO lines. This is not typical.
</polines>	
<gllines>	
<addbutton>>true</addbutton>	This required element is set to true by default. Set this to true if you want to give users the ability to add GL lines.
<linestoadd>1</linestoadd>	This element specifies the number of lines added to the eForm when the user clicks the Add Line button on the user interface.
<delbutton>>true</delbutton>	This required element is set to true by default. Set this to true if you want to give users the ability to delete GL lines.
<linewrapping>true</linewrapping>	<p>This element specifies whether GL code fields wrap to additional lines or remain displayed on a single line. Set to true to enable line wrapping of GL code fields.</p> <p>If line wrapping is enabled, you can define which line each field appears under the <glline label/> element.</p>
</gllines>	
<readonlyifexported>true</readonlyifexported>	This required element is set to true by default. It is recommended that this value always be set to true to prevent users from trying to export data more than one time, either by accident or intent. Setting this to true also prevents the user from modifying a previously exported invoice. The user receives an error message when attempting to export a second time.
<errorpopup>true</errorpopup>	This required element is set to true by default. Set this to true if you want the user to receive an error message containing a field validation error, such as an invalid date or amount. When set to false, the cursor stays on the input field until a valid value is entered with no error message.

Element	Description
<roles>	<p>Use these roles to configure the sections of the form that users and groups can view and modify in the form.</p> <p>You configure four user roles in the <roles> element. Specify one of the following values for the detail attribute of each user role, as outlined in the "About workflow queue roles" section:</p> <p>hidden: Use this option if the data is not displayed to the user in the form for that role.</p> <p>readonly: Use this option if the data is displayed but NOT modifiable by the user in the form.</p> <p>enabled: Use this option if the data is displayed and modifiable by the user in the form.</p>
<pre><role name="Invoice Entry" header="enabled" detail="hidden" po_detail=" hidden " gl_detail_po="hidden" gl_detail_nonpo=" hidden "></pre>	<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 values.</p>
<pre><queue invoicetype="nonpo" changetabs="true">APO Invoice Entry</queue></pre>	<p>The queue value in this entry and in ImageNow Workflow must match exactly. Add an entry for each workflow queue that performs 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>
</role>	
<pre><role name="Invoice Processing" header="enabled" detail="enabled" po_detail="enabled " gl_detail_po="enabled " gl_detail_nonpo="enabled "></pre>	<p>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.</p>
<pre><queue invoicetype="nonpo" changetabs="true">APO Invoice Processing</queue></pre>	<p>The queue value in this entry and in ImageNow must match exactly. Add an entry for each workflow queue that performs the PROCESSING 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>
</role>	

Element	Description
<pre><role name="Invoice Approval" header="readonly" detail="enabled" po_detail="enabled " gl_detail_po="enabled " gl_detail_nonpo="enabled "></pre>	<p>The Invoice Approval role is for users and groups who perform 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.</p>
<pre><queue invoicetype="nonpo" changetabs="true">APO Invoice Approval</queue></pre>	<p>The queue value in this entry and in ImageNow Workflow must be exactly the same name. Add an entry for each workflow queue that performs the APPROVAL 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>
<pre></role></pre>	
<pre><role name="(default)" header="readonly" detail="readonly" po_detail="enabled " gl_detail_po="enabled " gl_detail_nonpo="enabled "></pre>	<p>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.</p>
<pre></role></pre>	
<pre></roles></pre>	
<pre><fields></pre>	<p>The <fields> 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. Don't add to or delete from child elements in the <fields> element; instead, use the disable attribute. You should thoroughly test any changes you make to this element.</p>
<pre><header/></pre>	<p>The header label information includes mapped values that should not be changed. You can change the visible, width, disabled, maxlength, align, title, distributable, and the required attributes. Do not change the label or field attributes; instead, change the caption attribute.</p> <p>Notes</p> <p><distributable> is only applicable for the fields, Sales Tax, Freight Amount, Misc Amount, and the eight Additional Amount fields.</p> <p>If <distributable> 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>

Element	Description
<poline/>	The purchase order 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.
<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 <linewrapping> 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>
<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="lblHdr1" field="txtHdr1" visible="true captionkey="LBL_HDR_1:" 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.
<dupcheck_vsl enabled="true">[tab] =%s AND [f3] = %s</dupcheck_vsl>	<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 tab contains the Vendor ID and the document key F3 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) performs 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><statement view="All Documents" match="2" enabled="false">VendorID InvoiceNumber TotalAmount # 5 </statement></code>	<p>This element contains advanced logic for duplicate invoice check.</p> <p>The statement view attribute specifies the custom view that can be configured 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>Note 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<1 to 8>, Hdr<1 to 8></p>
<code></dupcheck></code>	

Element	Description
<polinematching>true</polinematching>	<p>This element specifies whether to perform PO matching at the line item level or at the header/footer level.</p> <p>A value of true in this element indicates that PO matching is performed at the line item level. In this case, iScript (AP_Validate_Server.js) matches PO lines from scanned invoices and locates the corresponding PO lines in the virtual table ("Z_APW_APPOLINE"). (Brainware is assumed.) Each line is matched by PO Item Number. If a match is found, the PO Line Number, PO Quantity, and PO Unit Cost from the virtual table are populated in the eForm. These fields are visible in the eForm, so the user can compare the values that are scanned and invoiced with the values that appear on the PO.</p> <p>If this value is set to true, set the AP_Skin_xml file to show the following elements (visible = true): txtPO_LineNumber, txtPO_Quantity, txtPO_UnitPrice, and imgPOLinesDialog. This check provides a dialog box for the user to look up PO lines in the virtual table and displays values that were matched from the PO lines in the virtual table.</p> <p>A value of false in this element indicates that PO matching is performed at the header and footer level. In this case, iScript (AP_Validate_Server.js) obtains the PO number from the scanned invoice, locates the PO lines from the virtual table ("Z_APW_APPOLINE") and loads the lines into the form. (Brainware is assumed.) From the form, the user can also manually perform this same function of loading PO lines from the virtual table.</p> <p>If this value is set to false, set the AP_Skin_xml file to hide the following elements (visible = false), as the following elements are not used: txtPO_LineNumber, txtPO_Quantity, txtPO_UnitPrice, and imgPOLinesDialog.</p>
<viewname>Z_APW All Projects</viewname>	The viewname element specifies the name of the custom view you created during the Accounts Payable Invoice eForm installation and setup process.
<scripts>	This element contains the scripts used by the form. For more information on each script, refer to the "inserver6\script" section in this document.
<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>

Element	Description
<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 5 = most verbose The default is 4.</p>
</AP_DupCheck>	
<AP_GetCompanies>	
<LogToFile type="boolean">>false</LogToFile>	<p>This element sets the logging for AP_GetCompanies.js to the console or to a log file.</p> <p>FALSE = the console TRUE = a log file The default is false.</p>
<DebugLevel type="int">4</DebugLevel>	<p>The logging level for AP_GetCompanies.js.</p> <p>0 = least verbose 5 = most verbose 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>	
<LogToFile type="boolean">>false</LogToFile>	<p>This element sets the logging for AP_GetData.js to the console or to a log file.</p> <p>FALSE = the console TRUE = a log file The default is false.</p>
<DebugLevel type="int">4</DebugLevel>	<p>The logging level for AP_GetData.js.</p> <p>0 = least verbose 5 = most verbose The default is 4.</p>

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

Element	Description
<MaxResults>1500</ MaxResults >	This element specifies the maximum GL codes to return at one time.
</AP_ GetGLCodeList>	
<AP_GetGLCodeUserConstraints>	
<LogToFile type="boolean">false</LogToFile>	<p>This element sets the logging for AP_GetGLCodeUserConstraints.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_GetGLCodeUserConstraints.js.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
<MaxResults>255</ MaxResults >	This element specifies the maximum constraint records to return at one time. This value cannot exceed 255.
</AP_ GetGLCodeUserConstraints >	
<AP_GetPOs>	
<LogToFile type="boolean">false</LogToFile>	<p>This element sets the logging for AP_GetPOs.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_GetPOs.js.</p> <p>0 = least verbose</p> <p>5 = most verbose</p> <p>The default is 4.</p>
<MaxResults>255</ MaxResults >	This element specifies the maximum purchase orders to return at one time. This value cannot exceed 255.
</AP_GetPOs>	
<AP_GetPOLines>	

Element	Description
<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.
<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.

Element	Description
<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.
<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 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 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 AP520 comment file.

Element	Description
<code><MA540FileName> MA540_%04d%02d%02d_%02d%02d%02d.csv </ MA540FileName ></code>	This element specifies the file name pattern to use for the MA540 file.
<code><VoucherBuildFileName> VoucherBuild_%04d%02d%02d_%02d%02d%02d.csv </ VoucherBuildFileName ></code>	This element specifies the File name pattern to use for the PeopleSoft Voucher Build file.
<code></AP_Invoice_Export></code>	
<code><AP_Maintenance></code>	
<code><LogToFile type="boolean">false</LogToFile></code>	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.
<code><DebugLevel type="int">4</DebugLevel></code>	The logging level for AP_Maintenance.js. 0 = least verbose 5 = most verbose The default is 4.
<code></AP_Maintenance></code>	
<code><AP_ReindexAndCPMapping></code>	
<code><LogToFile type="boolean">false</LogToFile></code>	This element sets the logging for AP_ReindexAndCPMapping.js to the console or to a log file. FALSE = the console TRUE = a log file The default is false.
<code><DebugLevel type="int">4</DebugLevel></code>	The logging level for AP_ReindexAndCPMapping.js. 0 = least verbose 5 = most verbose The default is 4.
<code></AP_ReindexAndCPMapping></code>	
<code><AP_Validate_Server></code>	

Element	Description
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_Validate_Server.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_Validate_Server.js. 0 = least verbose 5 = most verbose The default is 4.
<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>	
<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.

Element	Description
<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_VirtualTable_Installer.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_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>	

Element	Description
<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>	
<LogToFile type="boolean">false</LogToFile>	This element sets the logging for AP_VirtualTable_Updater.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_Updater.js. 0 = least verbose 5 = most verbose The default is 4.
<UpdaterConfigFile>AP_VirtualTable_Updater.xml</ UpdaterConfigFile >	This element specifies the file that contains the configuration information for updating virtual tables.
<DataFieldDelimiter>,</ DataFieldDelimiter >	This element specifies the delimiter to use in the input files to the VirtualTable_Updater.
<Connector>CSV</Connector>	This element specifies which connector to use to gather the input for updating the virtual tables.
</AP_VirtualTable_Updater>	
</scripts>	
</configuration>	

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="queue_name"	<p>The errorqueue attribute of the <configuration> 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 <configuration> 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 <readonlyifexported> 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 to route the workflow item to. 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
nextqueue="queue_name"	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 routeforward attribute to false if using a queue for this attribute.
outputdir="C:/inserver6/temp/ap/export"	<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>
<queues>	You can create a <queue> element for each workflow queue you need to export data from in this <queues> element. The value for the <queue> 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.
<queue>queue_name</queue>	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.
</queues>	
<worksheets>	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.
<worksheet>AP Invoice</worksheet>	
</worksheets>	
<invoice>	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.
<header>	

Element	Description
<field></field>	<p>Several attributes are available for each <field> element:</p> <p>Use the name attribute for documentation purposes.</p> <p>Use the sequence element for identifying the <field> elements and keeping track of how many are being used.</p> <p>Use the worksheet attribute to tie the field to a control on the form.</p> <p>Use the format attribute to set the format of the date for the type attribute when the type attribute value is set to date.</p> <p>Use the constant attribute for constant variables.</p> <p>Use the length attribute to set the length of the field and for validation in the form during data entry.</p> <p>Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.</p>
</header>	
<poline>	
<field></field>	<p>There are several attributes for each <field> element. Use the name attribute for documentation purposes. Use the sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. The format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.</p>
</poline>	
<glline>	
<field></field>	<p>There are several attributes for each <field> element. Use the name attribute for documentation purposes. Use the sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. The format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.</p>
</glline>	
</invoice>	

Element	Description
<nonpo>	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.
<header>	
<field></field>	There are several attributes for each <field> element. Use the name attribute for documentation purposes. Use the sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. The format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.
</header>	
<glline>	
</glline>	
</nonpo>	
<po>	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.
<header>	
<field></field>	There are several attributes for each <field> element. Use the name attribute for documentation purposes. The sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. The format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.
</header>	
<poline>	

Element	Description
<field></field>	There are several attributes for each <field> element. Use the name attribute for documentation purposes. Use the sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. The format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.
</poline>	
<glline>	
<field></field>	There are several attributes for each <field> element. Use the name attribute for documentation purposes. Use the sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. Use the format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.
</glline>	
<misc>	
<field></field>	There are several attributes for each <field> element. Use the name attribute for documentation purposes. Use the sequence element for identifying the <field> elements and keeping track of how many are being used. Use the worksheet attribute to tie the field to a control on the form. The format attribute is to set the format of the date for the type attribute when the type attribute value is set to date. Use the constant attribute for constant variables. Use the length attribute to set the length of the field and for validation in the form during data entry. Use the whole attribute to specify a whole number. Note that the length attribute should be equal to the sum of the whole and decimal attributes.
</misc>	
<po>	
</configuration>	

AP_Validate_Server.xml

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

Element	Description
<pre><configuration errorqueue="AP Error" worksheet="AP Invoice" pomatching="true" routeback="true"></pre>	<p>The configuration element attribute values in this file are used when unexpected validation errors occur during use of the form.</p> <p>The errorqueue attribute value is the name of the workflow queue where workflow items are located when an unexpected validation error occurs.</p> <p>The worksheet attribute is the name of the form. The default value is "AP Invoice".</p> <p>The pomatching attribute is set to either true or false. Set it to true if the form should import PO header line information from the business application or virtual tables. Otherwise, set it to false. You usually set this value to false when Brainware is capturing the data. This attribute does not apply when the form is a non-PO invoice. When the form is non-PO, the value in this setting is ignored.</p> <p>The routeback attribute controls whether the document routes back to the previous queue when validation errors occur in the AP_Validate_Server.js script.</p>
<datamappings>	
<pre><datamapping enabled="true" variance_calc="true" doctype="Invoice"></pre>	<p>This element contains the custom property and document key values used to synchronize with the form values. Set the enabled attribute to true to enable this synchronization. Set the variance_calc attribute to true if you are processing PO-based invoices and variance calculations are used by your workflow routing rules. You can set the enabled attribute to false to disable this feature or for testing purposes.</p> <p>Set doctype to Invoice or Credit Memo.</p>
<pre><customprops enabled="true"> <customprop/> <customprop/> </customprops></pre>	<p>These elements are set up by default when installed. Modify them as needed according to your indexing requirements.</p>
<pre><dockey enabled="true"> <dockey/> <dockey/> </dockey></pre>	<p>These elements are set up by default when installed and typically should be modified as needed according to your indexing requirements.</p>
</datamapping>	
</datamappings>	

Element	Description
<required_fields enabled="true">	Use this element to identify or configure the fields that are required to be input by the user in the related form section. These values are set with defaults during installation. Typically, these values do not require modification. However, your implementation might require the addition or removal of required fields. If so, you add or remove the field in the following child elements.
<pohdr ponumber="PONumber" enabled="true">	Invoice header fields for PO-based invoices
<worksheet enabled="true"></worksheet>	PO Number, Company, Invoice Number, Invoice Date, Total Amount
</pohdr>	
<nonpohdr enabled="true">	Invoice header fields for non-PO invoices
<worksheet enabled="true"></worksheet>	Company, Invoice Number, Invoice Date, Total Amount, Vendor ID, Tax Amount, Freight Amount.
</nonpohdr>	
<creditdr enabled="true">	Invoice header fields for credit based invoices
<worksheet enabled="true"></worksheet>	Invoice Number, Invoice Date, Total Amount, Invoice Reference Number, Credit Memo Reason
</creditdr>	
</required_fields>	
</configuration>	

AP_Custom_Cultures.xml

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

Element	Description
<cultures>	
<code><culture name="unique_name" base="language code-country/region code" language="language code"></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><language code> <language code>-<country/region code></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><numberformat groupseparator="separator_value" decimalseparator="separator_value" groupsizes="groupsize_values"></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><calendar separator="separator_value"></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><patterns shortinput="short_date" longinput="long_date" display="date"></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>Note The separator value for the display date should match the separator attribute value you specify in the calendar element.</p>
<code></calendar></code>	
<code><messages></code>	
<code><message key="field_name" caption="string"></code>	<p>The message element allows you to configure form label captions, error messages, and tooltips.</p> <p>The key attribute specifies a specific field in the form.</p> <p>The caption attribute specifies the label caption, error message, or tooltip value that you want to display for that field.</p> <p>Note For a complete list of form fields, see the Appendix: Form fields and string values section at the end of this document.</p>
<code></messages ></code>	
<code></culture></code>	
<code></cultures></code>	

AP_VirtualTable_Updater.xml

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

Element	Description
<pre><configuration inputpath="C:\inserver6\temp\ap\import*.csv" containsheaderline="true" archivepath="C:\inserver6\temp\ap\archive" errorpath="C:\inserver6\temp\ap\error"></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><recordstructure enabled="true"></pre>	
<pre><column enabled="true"></column></pre>	<p>The initial values for these elements are set up by default. These values do not need to be modified unless you are using a different structure in your CSV data files. The structure in this file needs to match the structure in your data files.</p> <p>There are several virtual tables and a corresponding <code><recordstructure></code> element for each virtual table. Some of the virtual tables are BUS_UNIT, GL_ACCT, PO, PO_LINE, and VENDOR.</p>
<pre></recordstructure></pre>	
<pre></configuration></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

If you use Brainware Distiller in conjunction with your form implementation, you do not use the Invoice Entry role. For more information refer to the AP eForm Configuration Guide.

About the reindex and custom 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 has been 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 difference is calculated as a variance amount and a variance factor and loaded 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 form is used must be listed in the <roles> section of the AP_Skin.xml file in the \inservice6\etc\ap directory. This file determines what role the form 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 "Modify the form 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) but could be used 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), but allow 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 is the default role 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 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: \$ can be a symbol that refers to the end of a line, but as a literal, it is a dollar sign. Some minor 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	<	This character should always appear in content as an escape character so that it does not interact with the syntax of the markup.
&	Ampersand	&	This character should always appear in content as an escape character so that it does not interact with the syntax of the markup.
'	Apostrophe	'	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 ' is part of the XML language, it is not defined in HTML. Use ' if text is passed to an HTML user agent.
"	Quotation mark	"	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. Note Use \" when writing unparsed data to XML.
%	Percent	%	The Notification Services XML vocabulary reserves the percent sign (%) for denoting parameters. Note \$#37 is an escape sequence, but not an entity reference defined by XML.
\	Slash	None required	Note 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.
gField5	The value of the document's Field5 document property.
gDrawer	The value of the document's drawer.

Variable	Description
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 perform 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.
- The following JavaScript events can be handled with custom code:

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.
onmouseup	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 onload 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.errorMessage = '" +
                "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 to be exported. 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 for 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.

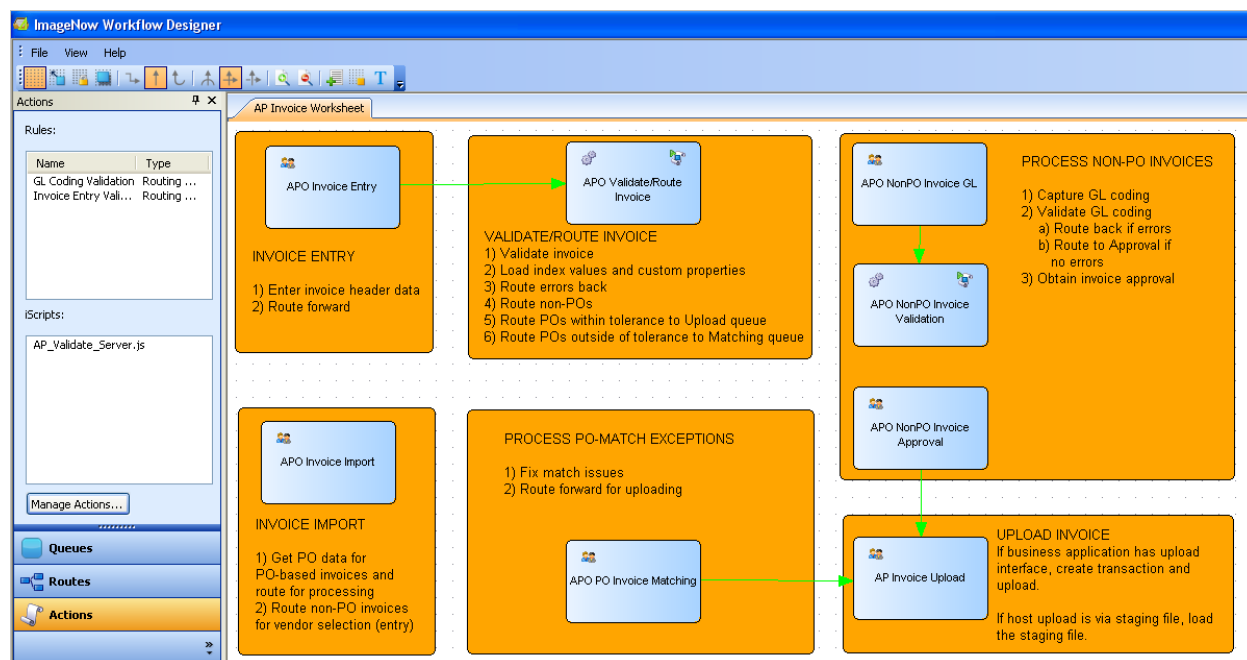
```
intool --cmd run-iscript --file AP_VirtualTable_Updater.js
```

4. After the command completes, perform 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 "Modify the form 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_DocType_Installer.js file, which was used in a previous procedure 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

In the workflow process containing the rule you want to modify, complete the following actions to create an Invoice Entry Validation routing rule:

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, perform the following substeps:
 1. In the **Rules name** box, type **Invoice Validation**.
 2. In the **Rule type** list, select **Routing rule**.
 3. Under **Statements**, perform the following procedures for each statement:
 - **Error Condition – Route Back**
 - **Non-PO Invoice**
 - **PO-based Invoice – Within Tolerance**
 - **PO-based Invoice – Outside of Tolerance**

Error Condition during Route Back

Complete the following steps to include a statement for Error Condition during Route Back:

1. In the **Rules Editor** dialog box, under **Statements**, click **New**.
2. In the **Statement Editor** dialog box, perform 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, perform 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, perform 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.

Non-PO Invoice

Complete the following steps to include a statement for Non-PO Invoice

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, perform 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, perform 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 **Field 5** 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, perform 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.

PO-based Invoice – Within Tolerance

Complete the following steps to include a statement for PO-based Invoice – Within Tolerance:

1. In the **Rules Editor** dialog box, under **Statements**, click **New**.
2. In the **Statement Editor** dialog box, perform 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, perform 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, perform the following substeps:
 1. In the **Business Process** list, select your workflow process.
 2. Under **Available queues**, select **AP 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.

PO-based Invoice – Outside of Tolerance

Complete the following steps to include a statement for PO-based Invoice – Outside of Tolerance:

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, perform 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, perform 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, perform 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, perform the following substeps:
 1. In the **Rules name** box, type **Invoice Validation**.
 2. In the **Rule type** list, select **Routing rule**.
 3. Under **Statements**, perform the following substeps to include a statement for **Check for Errors in GL Coding**:
 1. Click **New**.
 2. In the **Statement Editor** dialog box, perform 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, perform 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, perform 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

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 Country	Country Code	Supported Currency	Currency Code
Dutch	nl	Belgium Netherlands	BE NL	Euro Euro	EUR
English	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
English (en) is the default language.					
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 real	BRL
Spanish	es	Argentina Columbia Ecuador Mexico Spain	AR CO EC MX ES	Argentinian peso Columbian peso Ecuadorian sucre Mexican peso Euro	ARS COP ECS MXP 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 is configured to include 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	TableName	BusUnitID	CurrencyCode	CurrencyName	PayGroup	Active	
2	CURRENCY	1234	VEB	Venezuelan Bolívar 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	TableName	CurrencyCode	CountryCode	LocaleCode	Active	
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: Form fields and string values

The following table lists the fields and string values that you can configure in the AP_Custom_Cultures.xml.

Field	String Value
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
LBL_LOC	Location
LBL_PO_NUM	PO Number
LBL_PO_AMT	PO Amount
LBL_VEND_NAME	Vendor Name
LBL_VEND_ID	Vendor Number
LBL_REMIT_TO	Remit To
LBL_VEND_ADDR	Vendor Number
LBL_VEND_LOC	Vendor Location
LBL_VEND_SHORT_NAME	Short Name
LBL_VEND_PAY_TERMS	Payment Terms
LBL_VENDOR_VAT	VAT
LBL_VEND_FIELD_1	Field 1
LBL_VEND_FIELD_2	Field 2
LBL_INV_NUM	Invoice Number
LBL_INV_DATE	Invoice Date
LBL_INV_DUE_DATE	Invoice Due Date
LBL_INV_AMT	Invoice Amount

Field	String Value
LBL_INV_REF_NUM	Invoice Ref Num
LBL_CREDIT_REASON	Credit Reason
LBL_SALES_TAX	Sales Tax
LBL_FREIGHT	Freight
LBL_MISC_CHARGE	Misc.Charge
LBL_PAY_IMMEDIATELY	Pay Immediately
LBL_ADDL_AMT_1	Addl Amt 1
LBL_ADDL_AMT_2	Addl Amt 2
LBL_ADDL_AMT_3	Addl Amt 3
LBL_ADDL_AMT_4	Addl Amt 4
LBL_ADDL_AMT_5	Addl Amt 5
LBL_ADDL_AMT_6	Addl Amt 6
LBL_ADDL_AMT_7	Addl Amt 7
LBL_ADDL_AMT_8	Addl Amt 8
LBL_SUT_CODE	SUT Code
LBL_SUT_APPLY	SUT Apply
LBL_UUT_CODE	UUT Code
LBL_TAX_EXEMPT	Tax Exempt
LBL_CURRENCY	Currency
LBL_PAY_GROUP	Pay Group
LBL_SPECIAL_HANDLING	Special Handling
LBL_PAYMENT_COMMENTS	Payment Comments
LBL_HDR_1	Hdr1
LBL_HDR_2	Hdr2
LBL_HDR_3	Hdr3

Field	String Value
LBL_HDR_4	Hdr4
LBL_HDR_5	Hdr5
LBL_HDR_6	Hdr6
LBL_HDR_7	Hdr7
LBL_HDR_8	Hdr8
LBL_HDR_9	Hdr9
LBL_HDR_10	Hdr10
LBL_POLINE_MATCH_SELECT_ALL	Select All
LBL_POLINE_MATCH	Match
LBL_POLINE_TAXABLE_SELECT_ALL	Select All
LBL_POLINE_TAXABLE	Taxable
LBL_POLINE_NUM	Line #
LBL_POLINE_ITEM_NUM	Item #
LBL_POLINE_DESCRIPTION	Description
LBL_POLINE_QUANTITY	Quantity
LBL_POLINE_UOM	UOM
LBL_POLINE_UNITPRICE	Unit Price
LBL_POLINE_EXT_AMT	Extended Amount
LBL_POLINE_POLINE_NUM	PO Line #
LBL_POLINE_PO_QUANTITY	PO Quantity
LBL_POLINE_PO_UOM	PO UOM
LBL_POLINE_PO_UNIT_PRICE	PO Unit Price
LBL_POLINE_PO_1	PO1
LBL_POLINE_PO_2	PO2
LBL_POLINE_PO_3	PO3

Field	String Value
LBL_POLINE_PO_4	PO4
LBL_POLINE_PO_5	PO5
LBL_POLINE_PO_6	PO6
LBL_POLINE_PO_7	PO7
LBL_POLINE_PO_18	PO8
LBL_POLINE_PO_9	PO9
LBL_POLINE_PO_1 0	PO10
LBL_GLLINE_TAXABLE_SELECT_ALL	Select All
LBL_ADDL_ENTRY_DESC	Description
LBL_ADDL_ENTRY_AMT	Extended Amount
LBL_GLLINE_TAXABLE	Taxable
LBL_GLLINE_PCT	Pct
LBL_GLLINE_AMT	Amount
LBL_GLLINE_GL1	Account
LBL_GLLINE_GL1_DESC	Account Description
LBL_GLLINE_GL2	Unit
LBL_GLLINE_GL2_DESC	Unit Description
LBL_GLLINE_GL3	Sub Unit
LBL_GLLINE_GL3_DESC	Sub Unit Description
LBL_GLLINE_GL4	Sub Acct
LBL_GLLINE_GL4_DESC	Sub Acct Description
LBL_GLLINE_GL5	Activity
LBL_GLLINE_GL5_DESC	Activity Description
LBL_GLLINE_GL6	GL6
LBL_GLLINE_GL6_DESC	Activity Description

Field	String Value
LBL_GLLINE_GL7	GL7
LBL_GLLINE_GL7_DESC	Activity Description
LBL_GLLINE_GL8	GL8
LBL_GLLINE_GL8_DESC	Activity Description
LBL_GLLINE_GL9	GL9
LBL_GLLINE_GL9_DESC	Activity Description
LBL_GLLINE_GL10	GL10
LBL_GLLINE_GL10_DESC	Activity Description
LBL_GLLINE_GL11	GL11
LBL_GLLINE_GL11_DESC	Activity Description
LBL_GLLINE_GL12	GL12
LBL_GLLINE_GL12_DESC	Activity Description
LBL_GLLINE_GL13	GL13
LBL_GLLINE_GL13_DESC	Activity Description
LBL_GLLINE_GL14	GL14
LBL_GLLINE_GL14_DESC	Activity Description
LBL_GLLINE_GL15	GL15
LBL_GLLINE_GL15_DESC	Activity Description
LBL_GLLINE_GL16	GL16
LBL_GLLINE_GL16_DESC	Activity Description
LBL_GLLINE_GL17	GL17
LBL_GLLINE_GL17_DESC	Activity Description
LBL_GLLINE_GL18	GL18
LBL_GLLINE_GL18_DESC	Activity Description
LBL_GLLINE_GL19	GL19

Field	String Value
LBL_GLLINE_GL19_DESC	Activity Description
LBL_GLLINE_GL20	GL20
LBL_GLLINE_GL20_DESC	Activity Description
LBL_GLLINE_GL21	GL21
LBL_GLLINE_GL21_DESC	Activity Description
LBL_GLACCT_BUS_UNIT	Business Unit
LBL_GLACCT_ACCTG_UNIT	Acctg Unit
LBL_GLACCT_GLACCT	GL Acct
LBL_GLACCT_SUB_ACCT	Sub Account
LBL_GLACCT_ACCT_DESC	Account Description
LBL_GLACCT_5	GL Acct 5
LBL_GLACCT_6	GL Acct 6
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 #
LBL_CREDIT_DATE	Credit Date
LBL_CREDIT_AMT	Credit Amount
LBL_CUST_HDR_1	Cust HDR1
LBL_CUST_HDR_2	Cust HDR2

Field	String Value
LBL_CUST_HDR_3	Cust HDR3
LBL_CUST_HDR_4	Cust HDR4
LBL_CUST_HDR_5	Cust HDR5
LBL_CUST_HDR_6	Cust HDR6
LBL_CUST_HDR_7	Cust HDR7
LBL_CUST_HDR_8	Cust HDR8
LBL_CUST_HDR_9	Cust HDR9
LBL_CUST_HDR_10	Cust HDR10
TOOLTIP_DATE_FORMAT	Please enter date in {datemask1} or {datemask2} format.
TOOLTIP_CLOSE	Close
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_WINDOW	Look Up Account
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.

Field	String Value
ERR_PREVIOUSLY_EXPORTED	This invoice was previously processed.
ERR_REQUIRED_FIELDS	Required fields are missing.
ERR_GL_CODES_INVALID	GL codes are invalid.
ERR_SEARCH_LIMIT	incomplete
ERR_NO_MATCH	No match found.
ERR_VENDOR_NOT_FOUND	Vendor not found.
ERR_DUP_UNDEFINED	Duplicate check undefined in configuration file.
ERR_DUP_CHECK_MATCH	Duplicate check. Match count
ERR_DUP_CHECK	Duplicate check error. Check the log files for specifics.
ERR_DUP_INVALID_PARAM	Duplicate check error. Invalid parameter specified in config file.
ERR_PO_CLOSED	This PO is closed.
ERR_PO_NUM_NOT_FOUND	PO number not found.
CONFIRM_PO_NUMBER_CHANGE	Are you sure you want to change PO numbers? All existing lines will be replaced.
CONFIRM_PO_NUMBER_CHANGE_POLI NEMATCHING	Are you sure you want to change PO number? All existing PO information on lines will be cleared.
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
VendorNum	ID
VendorName	Vendor Name
Address	Address
RemitToCode	Remit To
MESSAGE_PO_NUMBER_CLEAR	To clear all PO and GL lines, press Clear Worksheet Data button above.

Field	String Value
LBL_MIN_CHAR	(minimum 3 characters)
LBL_OK	OK
LBL_CANCEL	Cancel
LBL_LINES_ADD	Add a line
LBL_LINES_DELETE	Delete this line
ERR_CONFIG	Configuration error: {value1} permission = '{value2}'. Changing to '{value3}'.
LBL_INVTYPE_PO	PO
LBL_INVTYPE_NON_PO	Non-PO
LBL_INVTYPE_CREDIT	Credit
LBL_ADDL_ENTRY_DESC	Description
LBL_ADDL_ENTRY_AMT	Extended amount
PO_GL_ERR_DIST	GL distributions no fully distributed

Index

AP workflow process.....	67	invoice header data.....	31
AP_CustomValidation_Client.js	62	iScript	68
AP_DocType_Installer.js.....	68	JavaScript	62
AP_GetData_Custom.js	63	JavaScript Event Handlers	62
AP_Validate_Server.js	58	Match Audit Exceptions	31
AP_VirtualTable_Updater.xml.....	18	PO matching	37
body onload event.....	62	Purge routine	25
body onunload event.....	63	reserved characters	60
Brainware	37, 53	routing actions.....	68
CSV data files	57	TableName	6
CSV files.....	66	user exits.....	62
custom logic	26	user roles	31
custom properties.....	20	validate.....	25
database lookups	62	validation.....	62
date format	34, 48	validation script	58
datemask.....	34	variance calculations	53
dbcall	63	Variance factor	58
dbcall method.....	63	Virtual Table CSV Files.....	6
dupcheck	35	Virtual Table Updater	18
Duplicate invoices	35	virtual tables.....	18, 19, 57
error queue.....	58	VSL query	35
incremental update.....	19	workflow queue roles	59
inputpath variable.....	27	WorksheetStylesheet.css	26
INTool.....	19, 48, 66, 75	XML reserved characters.....	60