Perceptive Intelligent Capture Connector for Accounts Payable Invoice eForm

Installation and Setup Guide

Version: 12.7.x

Written by: Product Knowledge, R&D
Date: August 2016

Lexmark™ Enterprise Software
formerly Perceptive Software

Table of Contents

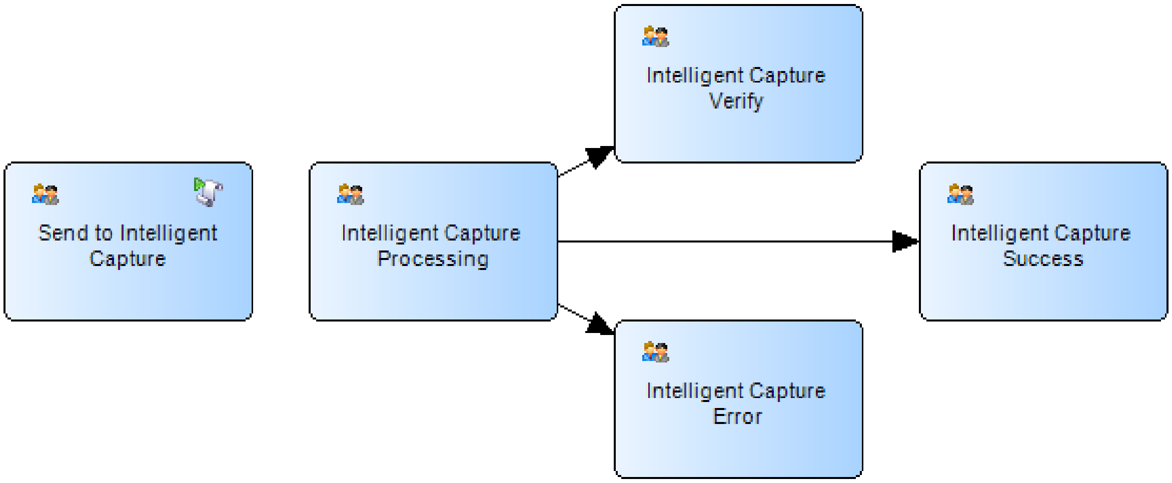
Product Overview

The Perceptive Intelligent Capture (PIC) Connector for AP Invoice eForm enables communication between Perceptive Content and Intelligent Capture for Invoices and allows Intelligent Capture to send status updates and extracted information to Perceptive Content. The Perceptive Content components that coordinate with Intelligent Capture using the PIC connector include Perceptive Content Workflow, iScripts, External Messaging Agent, and Output Agent.

The workflow process for an invoice includes the following procedures.

- The user captures an invoice into Perceptive Content using a capture profile. The profile captures the document into an initial workflow queue.
- The **Send to Intelligent Capture** queue contains the IC_ExportDoc.js iScript, which runs each time an invoice enters the queue. The iScript uses Output Agent to export the invoice as a TIFF image to an Intelligent Capture-monitored directory. After the export, Perceptive Content routes the original document to the Intelligent Capture Processing workflow queue.
- The Intelligent Capture Runtime service monitors and imports the invoice image from the directory. Intelligent Capture completes the Classification, Extraction, and Verification processes on the invoice image and generates one of the following external messages reflecting the invoice image status.
  - **Errors**
  - **Needs Verification**
  - **Success**
- The External Messaging Agent receives the external message through the connector and executes the IC_ReindexDoc.js iScript. The iScript routes the invoice, in workflow, to one of the corresponding queues.
  - **Intelligent Capture Success**
  - **Intelligent Capture Verify**
  - **Intelligent Capture Error**
- The iScript populates AP Invoice eForm with valid, extracted data and sets document and custom properties before routing to the **Intelligent Capture Success** queue. For this point in the process, you can set a routing rule to move the invoice to a queue for the next invoice-processing task.
- If the invoice needs verification, the user processes it with the Intelligent Capture Verifier. After the user reviews and validates the extracted data, Intelligent Capture sends a new external message. External Messaging Agent executes the iScript to route the invoice to either the **Intelligent Capture Success** or the **Intelligent Capture Error** queue.
- If the invoice has an error, the IC_ReindexDoc.js iScript populates AP Invoice eForm or custom eForm and sets index values with error information to aid in troubleshooting. For example, you can set a routing rule to handle certain conditions, such as VENDOR NOT FOUND which sends the document to a queue for the user to handle new vendors.

Below is an example of a primary workflow. Depending on your invoice processing needs, your process may vary.


Send to Intelligent Capture
Intelligent Capture Processing
Intelligent Capture Verify
Intelligent Capture Success
Intelligent Capture Error

Prepare to configure the solution

Solution prerequisites

Before downloading and installing the Intelligent Capture Connector for AP Invoice eForm, review and complete the following prerequisites, as necessary.

Program prerequisites

To install the Intelligent Capture Connector for AP Invoice eForm, you must install and have access to the following programs.

- Perceptive Content Server and Client 6.5.1 or higher
- Output Agent 6.5.1 or higher
- Intelligent Capture 5.2 or higher
- Intelligent Capture for Invoices 2.2 or higher
- AP Invoice eForm or custom eForm

For more information about the installation processes of each of these products, refer to guides available in the Product Documentation area of the Customer portal. For information on the compatibility of specific versions of the Perceptive Content and Intelligent Capture, refer to the Compatibility matrix for the complete solution.

Determine component versions

To determine the compatibility of Intelligent Capture for Invoices Connector, Intelligent Capture for Invoices, your existing ImageNow Server and AP Invoice eForm, review the following sections.

- Determine the Accounts Payable Invoice eForm and Intelligent Capture Connector version
- Determine the Perceptive Intelligent Capture for Invoices version
- Compatibility matrix for the complete solution

Determine the Accounts Payable Invoice eForm and Intelligent Capture Connector version

AP Invoice eForm and Intelligent Capture Connector release concurrently, and use the same version and build number. If you have previously installed AP Invoice eForm and need the existing version number, complete the following steps.

1. Navigate to **[*drive*:]\inserver\<*form or worksheet*>\data_definition** and open **AP_Invoice.xml** with a text editor.
2. In the top five lines of text, locate the version number.

Note If no version number exists, you have an AP Invoice eForm version prior to 10.0.0.28. Contact your Perceptive Software representative to determine the exact version number.

Determine the Perceptive Intelligent Capture for Invoices version

If you have previously installed Perceptive Intelligent Capture for Invoices and need the existing version number, complete the following steps.

1. Navigate to **[*drive*:]\Program Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\**, right-click on **BW Invoices.sdp**, and click **Properties**.
2. On the **Custom** tab, in the **Properties** box, in the **Name** column, locate **PS_DFI_VER** and note the version number in the **Value** column.
3. Open **BW Invoices.sdp** in **Intelligent Capture Designer**.
4. Click the **Switch to Definition Mode** button.
5. In the left pane, on the **Classes** tab, right-click **Invoices** and click **Show Script**.
6. In the **Script View for Document Class “Invoices”** window, in the top 10 lines of the script, in the comments, locate the build number next to **BUILD**.

Compatibility matrix for the complete solution

The following table maps the versions of ImageNow server, AP Invoice eForm, Intelligent Capture for Invoices, and Intelligent Capture that are compatible with the Intelligent Capture Connector for AP Invoice eForm.

Important Due to shared library files, you must use the same version and build number for AP Invoice eForm and Intelligent Capture Connector. When upgrading, update the Intelligent Capture Connector for AP Invoice eForm first and then the AP Invoice eForm. Keep AP Invoice eForm and Intelligent Capture Connector for AP Invoice eForm on the same build number.

For versions of AP Invoice eForm and Intelligent Capture Connector for AP Invoice eForm prior to 10.0.0.28, contact your Perceptive Software representative to determine compatibility.

Perceptive Content (ImageNow) Server	**Perceptive AP Invoice eForm and Perceptive Intelligent Capture Connector for AP Invoice eForm**
6.5.1.149 to 6.6.x	10.0.0.28 to 12.7.x*
6.7.x	10.1.0.307 to 12.7.x
7.x.x	12.1.0.169 to 12.7.x

Note

- *Intelligent Capture Connector and Perceptive AP Invoice eForm 12.7.x, when used with PerceptiveConnector for SAP Financials, is compatible with ImageNow 6.7.x or higher.
- Perceptive AP Invoice eForm and Intelligent Capture Connector for AP Invoice eForm share common library files and must remain on the same version.

Install PIC Connector for AP Invoice eForm

To install Intelligent Capture Connector for AP Invoice eForm, complete the following tasks.

- Extract and load the connector scripts
- Copy the XML files

Extract and load the connector scripts

To install the connector, complete the following steps.

1. Navigate to the **Perceptive Intelligent Capture Connector for AP Invoice eForm 12.7.0 (<*Build number*>).zip** file you previously downloaded.
2. Extract the **\inserver\script** folder to the **[*drive*:]\inserver\script** directory.

   Note If you installed AP Invoice eForm in an active-active server environment, extract to the **[*shared path*]\inserver\script** directory.

   This folder loads the following iScripts.

   - **IC_WS_Mapping.js** contains logic specifically for mapping Intelligent Capture Invoice XML to AP Invoice eForm.
   - **IC_Custom_WS_Mapping.js** contains user exits to script custom logic for mapping Intelligent Capture XML to a custom eForm.

     Note When you upgrade the connector, replace this file with your backup copy that contains your custom scripting.
   - **IC_ExportDoc.js** exports invoice images to Intelligent Capture.

- **IC_ReindexDoc.js** indexes invoice images and updates the AP Invoice eForm or custom eForm data from Intelligent Capture.
- **IC_LI_Review.js** identifies invoices in workflow with match discrepancies and line items that do not match the PO by Intelligent Capture.

Copy the XML files

To copy the connector XML files, complete the following step.

- Copy the contents of the **Perceptive Intelligent Capture Connector for AP Invoice eForm 12.7.0 (<*Build number*>).zip\inserver\etc\ap** folder and place it in the **[*drive*:]\inserver\etc\AP** folder.

  Note If you installed AP Invoice eForm in an active-active server environment, extract to the **[*shared path*]\inserver\etc\ap** directory.

  This action loads the following XML files.

  - **AP_VirtualTable_Updater.xml** provides an example of how to configure the virtual table updater to export configured data automatically to a CSV file for Intelligent Capture consumption. For example, this file can send business unit or company code information from your virtual tables to Intelligent Capture.
  - **IC_Config.xml** controls the basic functions of Intelligent Capture Connector.

Upgrade PIC Connector for AP Invoice eForm

To upgrade PIC Connector for AP Invoice eForm from a version lower than 12.7.0, complete the following steps.

1. Back up the existing files in the **[*drive:*]\inserver\etc**, **\script**, and **\form** or **\worksheet** directories to a local directory before installing the latest connector scripts.
2. Navigate to the **Perceptive Intelligent Capture Connector for AP Invoice eForm 12.7.0 (<*Build number*>).zip** file you previously downloaded.
3. Extract the ZIP file to a local directory.
4. Copy the following files from the extracted **[*drive:*]\{path}\script** directory to the **[*drive*:]\inserver\script** directory to replace the existing files.
   - **IC_WS_Mapping.js**
   - **IC_ExportDoc.js**
   - **IC_ReindexDoc.js**
   - **IC_LI_Review.js**
5. Copy the IC_OperationLib.jsh file from the extracted **[*drive:*]\{path}\script\apef\includes** directory.
6. Copy the **IC_Config.xml** file from the extracted **[*drive:*]\{path}\etc\ap** directory to the **[*drive:*]\inserver\etc\ap** directory.

   Note After you replace the previous version of **IC_Config.xml** with the latest version, open the backup of **IC_Config.xml** and copy the existing configurations to the latest version.
7. Open the backup of **IC_ReindexDoc.js** and **IC_ExportDoc.js** files.
8. Open the **IC_Config.xml** file from the **[*drive:*]\inserver\etc** directory.

9. From the backup of **IC_ReindexDoc.js** file, copy the values of the configuration parameters to the following parameters under `<IC_ReindexDoc>` in the **IC_Config.xml** file.
    - `logtofile`
    - `debuglevel`
    - `lineitemdelimiter`
    - `lineitemouterdelimiter`
    - `propdocumentid`
    - `propstatuscode`
    - `propreasoncode`
    - `icdoctypecredit`
    - `indoctypecredit`
    - `icdoctypeinvoice`
    - `indoctypeinvoice`

    Note In IC_ReindexDoc.js, the configuration parameters BW_DOCTYPE_CREDIT and BW_DOCTYPE_INVOICE are renamed as icdoctypecredit and icdoctypeinvoice in IC_Config.xml, respectively. The default value for PROP_STATUS_CODE in IC_ReindexDoc.js, bw_status, is changed to ic_status in IC_Config.xml. Also, the configuration parameters ERR_DUP_CHECK and CONFIG_VERIFIED are not valid any more.
10. From the **IC_ExportDoc.js** file, copy the values of the configuration parameters to the following parameters under `<IC_ExportDoc>` in the **IC_Config.xml** file.
    - `logtofile`
    - `debuglevel`
    - `icholdingq`
    - `errorq`
    - `defaultseparator`

    Note In IC_ExportDoc.js, the configuration parameter `BW_HOLDING_Q` is renamed to `icholdingq` in IC_Config.xml. Also, `the configuration parameter CONFIG_VERIFIED is not valid any more.`

Configure PIC Connector for AP Invoice eForm

To configure the Intelligent Capture Connector scripts, complete the following procedures.

- Configure Perceptive Intelligent Capture for Invoices User Exit scripts
- Configure Output Agent
- Configure External Messaging Agent
- Configure Perceptive Content workflow
- Configure connector features: IC_Config.xml
- Configure value added tax (VAT) features

Important Before configuring the scripts, activate any changes to the initialization file by restarting Perceptive Intelligent Capture services.

Configure Perceptive Intelligent Capture for Invoices User Exit scripts

To modify the Intelligent Capture for Invoices script, complete the following steps.

1. Open the project file (.sdp file) with **Intelligent Capture Designer**.
2. Click the **Definition Mode** button.
3. Open the **Invoices** tree.
4. Click the **Show/Hide Scripts** button.
5. To display global variables, click the **2** tab on the left side of the window. Complete the following substeps to configure the Perceptive Content connection string.
    1. Using the **Edit/Find** option, search for **ImageNow**. The system returns the ImageNow connection string.
    2. To stop the scripting module, in the **Script View** toolbar, click the **End** button.
    3. Modify the **Data Source** value with the name of the ImageNow Server computer. For example,

       ```
       Info=True;User ID=inemuser;Initial Catalog=INOW6;Data Source=Server01”
       ```
    4. Locate the dbType connection string and modify as needed to **Oracle** or **SQLServer**. For example,

       ```
       Global Const dbType As String = "Oracle"
       ```
    5. Locate the INOW connection string and modify as needed. For example,

       ```
       Global Const INowConnectionString As String =
       "Provider=OraOLEDB.Oracle.1;Password=imagenow;Persist Security Info=True;User
       ID=inuser;Data Source=INOW"
       ```
6. To restart the scripting module, in the **Script View** toolbar, click the **Start/Resume** button.
7. Exit the **Script View** function.
8. To save your changes, in the **Designer** toolbar, click the **Save** button.

Configure Output Agent

To configure Output Agent, complete the following steps.

1. Navigate to **[*drive*:]\inserver\etc** folder and open the **inserveroutput.ini** file in a text editor.

   Note If working in an active-active server environment, navigate to **[*shared path*]\inserver\etc**.
2. Under **[Monitor]**, set the **poll.interval**. For example, set the interval to **1** or **2** seconds.
3. Set **export.multipage.document** to **TRUE**.

Configure External Messaging Agent

To configure External Messaging Agent, complete the following steps.

1. Navigate to the **[*drive*:]\inserver\etc** directory and open the **inserverEM.ini** file with a text editor.

   Note If working in an active-active server environment, navigate to **[*shared path*]\inserver\etc**.

2. Insert the following lines into the INI file.

   ```
   [brainware]
   bw_status_update=IC_ReindexDoc.js
   ```

   Note If you use a custom eForm with Intelligent Capture, the script name you use here must be the custom script used to complete the importing and indexing into Perceptive Content.

Configure Perceptive Content workflow

Define a workflow process with queue names that indicate the invoice status. Example queue names include the following.

- Send to Intelligent Capture
- Intelligent Capture Processing
- Intelligent Capture Success
- Intelligent Capture Verify
- Intelligent Capture Error

Note If your solution implementation incorporates Mail Agent Conversion (INMAC), the conversion should take place after the “success” queue.

Configure the name of each process queue in the scripts. After the invoice reaches the success queue, a routing rule must route the invoice to the beginning of the invoice workflow. In this queue, a script runs to fill in the data Perceptive Content requires, and then sends the document to validation like any other invoice.

Important Add the following script actions and routing rules to the invoice workflow.

- In the **Send to Intelligent Capture** queue, add **IC_ExportDoc.js** as an inbound action.
- In the **Intelligent Capture Success** queue, add the following routing rule.

  ```
  If F5 = blanks (meaning NonPO), execute the same routing rule currently assigned to
  the validation queue in your AP invoice processing workflow for NonPO invoices.

  If F5 is NOT Blank (meaning PO), execute the same routing rule currently assigned
  to the validation queue in your AP invoice processing workflow for PO invoices. You
  do not have to run invoices from Intelligent Capture through the Validation Server
  script.
  ```

- In the **Intelligent Capture Error** queue, add the following routing rule.

  ```
  If F5 = blanks (meaning NonPO) and AP Custom Valid (CP) starts with "Vendor", route
  to the New Vendor Setup queue in your AP Invoice Processing workflow.

  If F5 = blanks (meaning NonPO) and AP Custom Valid (CP) is not blank (meaning some
  other error has occurred), route to the queue designated for NonPO invoice errors.

  If F5 is NOT Blank (meaning PO) and AP Custom Valid (CP) is not blank (meaning some
  other error has occurred), route to the queue designated for PO invoice errors.
  ```

Important To separate invoices automatically with match discrepancies, you need to use the workflow script IC_LI_Review.

- Add **IC_LI_Review.js** as an inbound action to a workflow queue, and then add the following routing rule.

```
If AP Detail Valid (CP) starts with "IC_LI_REVIEW" (this is an indication of a
severe error in the script), route the invoice to a special exception queue.

If AP Detail Valid (CP) starts with “ERR-MATCH” (this is an indication that the
invoice has line items that need manual pairing and/or there is match
discrepancies), route the invoice to the workflow queue where match discrepancies
and/or manual pairing are to be handled.

All else – route to the next step in the process.
```

Configure connector features: IC_Config.xml

The IC_Config.xml file controls the basic functions of Intelligent Capture Connector and the IC_ReindexDoc.js and IC_ExportDoc.js scripts.

If you upgrade the connector, use your backup copy of IC_Config.xml to configure the latest version of this file with your values.

The following sections describe the configuration options for IC_Config.xml.

Lookup validation

If the `enable` attribute of the `<validation>` element is set to `true`, IC_Config.xml compares the data captured from the invoice with the data extracted using lookup from the configured data source for AP Invoice eForm.

```
<validation enable="true">
```

The validation checks for information that is missing or does not match. Any information that Intelligent Capture is unable to capture from the invoice is automatically completed by the lookup data. Any data that does not match is either replaced or an appropriate error message displays.

Replace vendor number

If a mismatch in vendor numbers is found during validation, you can configure the IC_Config.xml file to replace the vendor number captured from the invoice with the vendor number extracted using lookup. To enable vendor number replacement, set the value of <forcevendormatch> element to `true`.

```
<forcevendormatch>true</forcevendormatch>
```

If the value is set to `false`, the vendor number captured from the invoice is not replaced.

Configure separator for multiple PO

The PIC for Invoices connector supports extraction of multiple PO numbers and displays them as a concatenated list with a delimiter. You can specify the delimiter under the <poseparator> tag.

```
<poseparator>$</poseparator>
```

By default, a `;` (semi-colon) is the delimiter.

Intelligent Capture export and error directories

This element specifies the directory where Intelligent Capture places processed (exported) invoices.

```
<importdir>C:\\Program
Files\\Perceptive\\Projects\\PerceptiveInvoiceSolution\\Export\\</importdir>
```

This element specifies the directory where Intelligent Capture sends XML files and images when an error occurs.

```
<errordir>C:\\Program
Files\\Perceptive\\Projects\\PerceptiveInvoiceSolution\\Error\\</errordir>
```

Error queues

This optional error queue value specifies the workflow queue that receives invoices if an operational failure occurs.

```
<errorqueuereindex>Intelligent Capture Error</errorqueuereindex>
```

This optional error queue value specifies the workflow queue that receives invoices if an error occurs in the IC_ExportDoc process.

```
<errorqueueexport>Intelligent Capture Error</errorqueueexport>
```

Document notes for invalid reason codes

By default, invalid reason codes only load into the AP Custom Valid custom property. Therefore, <setdocnotes> is set to FALSE.

```
<setdocnotes>false</setdocnotes>
```

If you set it to TRUE, Intelligent Capture loads invalid reason codes into both the AP Custom Valid custom property and the Document Notes section.

Date mask

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.

```
<datemask>MMDDYYYY</datemask>
```

Custom valid property

This element specifies the custom property name for storing the Intelligent Capture Invalid Reasons value.

```
<customvalidprop>AP Custom Valid</customvalidprop>
```

Intelligent Capture XML data storage

The subobject element allows you to store the complete set of XML data exported by Intelligent Capture. If enabled, the IC_ReindexDoc.js script stores the XML as a DCExport subobject of the Perceptive Content document. To enable, set to true. The default is false.

```
<subobject enable="true"/>
```

To provide a user-friendly version of this XML in the latest IC_ReindexDoc.log file, set the debug level in IC_ReindexDoc.js to 5 (DEBUG).

IC_ReindexDoc.js uses the Intelligent Capture XML root node as the subobject `[workingName]` property. For example, if the name of the default root node is `<BrainwareDocument>`, then the subobject `[workingName]` value is `BrainwareDocument`.

Each Perceptive Content document can have one DCExport subobject per unique workingName. For example, a document processed with two Intelligent Capture projects, such as Mailroom and Invoices, could have two DCExport subobjects. However, a document processed twice with the Invoices project would only retain the latest data as the subobject.

The DCExport subobject is also available through iScript. For information on one way to retrieve the data with iScript, refer to the getSubObject method in the **INVersion** topic in ImageNow 6.7.x online help.

Workflow queue names

You can configure the workflow queue names Intelligent Capture uses. Do not change the status code settings unless you are making corresponding changes in Intelligent Capture.

```
<statuscode code="800" type="SUCCESS" queuename="Intelligent Capture Success"/>
<statuscode code="550" type="VERIFICATION" queuename="Intelligent Capture Verify"/>
<statuscode code="DEFAULT" type="ERROR" queuename="Intelligent Capture Error"/>
```

Document type mapping

You can configure the document type mapping Intelligent Capture uses.

```
<doctype name="Invoice" bwimportdir="C:\\Program
Files\\Perceptive\\Projects\\PerceptiveInvoiceSolution\\Import\\">
//Defines the ImageNow document type and the directory where ImageNow images are moved
for Intelligent Capture processing.
```

Value added tax (VAT) validation

You can configure this section to enable or disable the **currency** and **VAT summary total** features. For more information on VAT, refer to Configure value added tax (VAT) feature.

```
<validation>
   <currency>false</currency>
   <vatsummarytotal>false</vatsummarytotal>
</validation>
```

Batch separation

You can configure the IC_Config.xml file to export document properties as part of the file name. You can also export files to multiple Intelligent Capture import directories. You must include the <batchseparation> node for each document type that uses the batch separation functionality.

The separator attribute defines what character to use to separate the different identifier attributes in the exported image filename. Valid characters include ^, #, +, and ~.

The identifier attribute determines the batch separation and must be a variable. The variable's value should match the name attribute of an <identifier> node. This determines which <identifier> node to use for separation.

For each **<identifier>** node, define the following attributes.

- **name**
- **groupid**. In DFI.ini, in the IMP_VL_ClientID setting, this provides the value for COMPONENT2.

- **batchid**. In DFI.ini, in the IMP_VL_ExternalBatchID setting, this provides the value for COMPONENT3.
- **transid**. In DFI.ini, in the IMP_VL_TransactionID setting, this provides the value for COMPONENT4.
- **transtype**. In DFI.ini, in the IMP_VL_TransactionType setting, this provides the value for COMPONENT5.
- **bwimportdir**

```
<doctype name="Invoice" bwimportdir=" C:\\Program
       Files\\Perceptive\\Projects\\PerceptiveInvoiceSolution\\Import\\">
       <batchseparation separator="^" identifier="(currentQueue)">
       <identifier name="Send to Queue A" groupid="{Vendor ID}" batchid="Batch ID"
       transid="Trans ID" transtype="<any alpha-numeric string>"
       bwimportdir=" C:\\Program
       Files\\Perceptive\\Projects\\PerceptiveInvoiceSolution\Import\\"/>
      <identifier name="Send to Queue B” groupid=”{Invoice Number}" batchid="Batch ID"
       transid="3" transtype="4" bwimportdir=”C:\\Brainware
       Projects\\AP\\Import\\B\\"/>
   </batchseparation>
```

Notes

- The value of groupid must be numeric. The values of batchid, transid, and transtype can be alpha numeric.
- name and bwimportdir must be an alphanumeric string
- batchid, transid, and transtype support empty string values
- groupid must always be populated

You can use the following variables.

- (previousQueue), (currentQueue)
- Index keys in the format [index key]. Use any of the following index keys: [docid], [drawer], [folder], [tab], [field1], [field2], [field3], [field4], [field5], [doctype], [creationusername], and [docname]. Only ImageNow 6.7 and higher support [docname].
- Custom properties in the format {Custom Property}.

Duplicate invoice detection

Duplicate invoice detection allows Intelligent Capture connector for AP Invoice eForm to identify duplicate invoices within Perceptive Content and process them based on how you configure IC_Config.xml file. You can configure the IC_Config.xml file to identify duplicate invoices using either of the following process.

- Legacy invoice check
- Advanced duplicate check

Legacy invoice check

The legacy invoice check compares the Intelligent Capture-extracted vendor ID and invoice number to specified Perceptive Content fields. You can restrict the search to specific document types.

To configure the IC_Config.xml file to include a VSL query statement, complete the following steps.

1. In **IC_Config.xml**, verify that `<dupcheck>` is set to `"true"`.
2. Below `<dupcheck>`, specify the XPath to the vendor ID and invoice number data in Intelligent Capture.

   Example

   ```
   <vendorid>/BrainwareDocument/InvHeader/VENDORID</vendorid>
   <invoicenumber>/BrainwareDocument/InvHeader/INVOICENO</invoicenumber>
   ```

3. In the `<dupcheck_vsl>` element, specify the Perceptive Content keys or custom properties that contain the **vendor ID** and **invoice number**, in that order.

   Valid options include [folder], [tab], [field1], [field2], [field3], [field4], [field5], and custom properties in the form {custom property}.

   Example

   ```
   <dupcheck_vsl>{Vendor ID} = %s and [f4] = %s </dupcheck_vsl>
   ```

4. Optional. To restrict the search by document type, set `<dupcheck_vsl>` with `[DocType] = '<document type>'`.

   Example

   ```
   <dupcheck_vsl>{Vendor ID} = %s and [f4] = %s
   and [DocType] = 'Invoice'</dupcheck_vsl>
   ```

5. To send an error message to a document key or custom property, configure **<customprop>** with the key or custom property and a value for the error message. This alerts users to duplicate invoice issues and allows you to create routing rules based on the value of the property.

Example 1

```
<dupcheck value="true">
      <vendorid>/BrainwareDocument/InvHeader/VENDORID</vendorid>
      <invoicenumber>/BrainwareDocument/InvHeader/INVOICENO</invoicenumber>
      <dupcheck_vsl>[tab] = %s AND [f3] = %s</dupcheck_vsl>
      <customprop cp="AP Header Valid" value="ERR_DUP_INVOICE"/>
</dupcheck>
```

Example 2

```
<dupcheck value="true">
      <vendorid>/BrainwareDocument/InvHeader/VENDORID</vendorid>
      <invoicenumber>/BrainwareDocument/InvHeader/INVOICENO</invoicenumber>
      <dupcheck_vsl>{Vendor ID} = %s and [f4] = %s
      and [DocType] = 'Invoice'</dupcheck_vsl>
</dupcheck>
```

Advanced duplicate check

The advanced invoice check searches for duplicate invoices based on a flexible set of eForm fields. For example, you can set the query to look at vendor ID, invoice number, and invoice date. You set how many of those parameters must match to indicate a duplicate. You can also specify a tolerance level for fields that contain monetary values.

To configure IC_Config.xml with the advanced duplicate invoice check, complete the following steps.

1. In **IC_Config.xml**, verify that `<dupcheck>` is set to `"true"`.
2. Remove or comment-out the existing legacy duplicate check nodes `<vendorid>`, `<invoicenumber>` and `<dupcheck_vsl>`.
3. Remove the comment characters from the `<statement>` element and complete the following substeps.
   - To filter the statement query with an Perceptive Content view, set **view** with a view name.

     Note The default view is "All Documents". You can create a custom view definition within Management Console and specify that view name here.
   - To set the number of matching parameters that indicate a duplicate invoice, set **match** with a whole integer, such as 2.
4. Set the parameters, with a **|** (pipe character) and a space on each side as the separator.

   The parameters correspond to the worksheet attributes in the AP_Export.xml file. The valid parameters are CompanyNumHdr, LocationCode, PONumber, VendorID, InvoiceNumber, InvoiceDate, TotalAmount, InvoiceReferenceNumber, TaxAmount, FreightAmount, MiscCharge, AddlAmt[1-8], and Hdr1[1-8].

   To set a variance percentage for a monetary value, after the parameter, add `%` and an integer. For example, to allow a variance of 1.5% on the Tax Amount field, use the following code:

   ```
   TaxAmount % 1.5
   ```

   To set a variance amount for a monetary value, after the parameter, add # and an integer. For example, to allow a variance of five units, such as dollars, on the Total Amount field, use the following code:

   ```
   TotalAmount # 5
   ```

In the following example, the query compares the vendor ID, invoice number, and total amount of the current invoice against all other invoices available through the AP Documents view. If another invoice has two or more parameters that match, Intelligent Capture flags the current invoice as a duplicate.

Example

```
<dupcheck value="true">
<statement view="AP Documents" match="2">VENDORID | INVOICENUMBER | TotalAmount #
5</statement>
</dupcheck>
```

Reindex duplicate invoices

After IC_ReindexDoc.js identifies a duplicate invoice, it can either reindex the document keys or not reindex them. To configure IC_Config.xml to reindex duplicate invoices, complete the following steps.

1. Open the **IC_Config.xml** file.
2. Set the `allowReindexingOfDuplicates` attribute of the `<dupcheck>` element to `"true"`.

Note You can reindex document keys of duplicate invoices for both the methods – Legacy invoice check and Advanced duplicate check. By default, the value for `allowReindexingOfDuplicates` attribute is set to `"false"`.

Important Before you enable reindexing of document keys of duplicate invoices, ensure that a unique document key exists to avoid merging of the duplicate invoice to the original invoice.

Index Mapping

The default index key mappings are preconfigured to work with AP Invoice eForm. If you work with a custom eForm, you may need to adjust the mappings to coincide with your specific indexing requirements. For each index key mapping row, fill in values for the **key** and optional **function** and **mapping** attributes.

- **key**. The Perceptive Content document key. Valid values are **folder**, **tab**, **f1**, **f2**, **f3**, **f4**, **f5**.

  Note The capture profile sets the drawer and document type values.
- **function**. An optional attribute that lets you include a java function such as **to lowercase** or **round**.
- **mapping**. An optional attribute that converts values from Intelligent Capture to your eForm, such as `mapping={'PO' : 'po', 'NO-PO' : 'non-po'}`.

Example

```
<indexkey xpath="/BrainwareDocument/InvHeader/COMPANYCODE" key="folder" function=""
mapping="{'1' : '001'}"/>
<indexkey xpath="/BrainwareDocument/InvHeader/VENDORID" key="tab" function=""
mapping=""/>
<indexkey xpath="/BrainwareDocument/InvHeader/INVOICENO" key="f3" function=""
mapping=""/>
<indexkey xpath="/BrainwareDocument/InvHeader/PO" key="f5" function="" mapping=""/>
```

Custom property mapping

The custom property settings in the script are preconfigured to work with AP Invoice eForm. If you are working with a custom eForm, you may need to adjust the following settings to satisfy your requirements.

- **cp**. The custom property name.
- **datatype**. Use **"STRING"** for alphanumeric or date values or **"NUMBER"** for numeric values. The default setting is "STRING".
- **default**. The default value if the corresponding value from Intelligent Capture is null.
- **function**. An optional attribute that lets you include a Java function such as **to lowercase** or **round**.
- **mapping**. An optional attribute that converts values from Intelligent Capture to your eForm, such as `mapping={'PO' : 'po', 'NO-PO' : 'non-po'}`.

Example

```
<customprop xpath="/BrainwareDocument/InvHeader/COMPANYCODE" cp="Company"
datatype="STRING" default="" function="" mapping="" />
<customprop xpath="/BrainwareDocument/InvHeader/VENDORID" cp="Vendor ID"
datatype="STRING" default="" function="" mapping="" />
<customprop xpath="/BrainwareDocument/InvHeader/VENDORNAME” cp="Vendor Name”
datatype="STRING" default="" function="" mapping="" />
<customprop xpath="/BrainwareDocument/InvHeader/INVOICENO" cp="Invoice Number"
datatype="STRING” default="" function="" mapping="" />
<customprop xpath="/BrainwareDocument/InvHeader/AMOUNT" cp="Invoice Amount"
datatype="NUMBER" default="" function="" mapping="" />
<customprop xpath="/BrainwareDocument/InvHeader/INVOICEDATE" cp="Invoice Date”
datatype="DATE" default="" function="" mapping="" />
<customprop xpath="/ BrainwareDocument/InvHeader/TAX" cp="Invoice Tax Amount"
datatype="NUMBER" default="" function="" mapping="" />
<customprop xpath="/BrainwareDocument/InvHeader/PO" cp="PO Number" datatype="STRING"
default="" function="" mapping="" />
```

eForm name and data definition

This setting specifies the eForm name and the path for the data definition. The default setting is preconfigured to work with AP Invoice eForm. If you work with a custom eForm, you may need to configure this setting.

Example

```
<worksheet name="AP Invoice"
datadef="C:\\inserver\\form\\data_definition\\AP_Invoice.xml">
```

PO line items

This element allows you to enable and disable Intelligent Capture PO line item processing. If set to FALSE, the virtual tables provide the data used to populate the AP Invoice eForm line item section. If set to TRUE, Intelligent Capture provides the data.

The default is FALSE.

Example

```
<functiontoapply name="doAPInvoiceMapping">
      <polineitemprocessing>false</polineitemprocessing>
</functiontoapply>
```

For more information about virtual tables, refer to the *Perceptive AP Invoice eForm Connector Configuration Guide*.

Location code

Use this setting to define the source of the Location code, which is provided at scan time. Intelligent Capture does not provide location code. If using a location code, this value must pass to the Intelligent Capture Connector using an index value that the capture profile sets at scan time. By default, Intelligent Capture uses the Perceptive Content document key F4 (field4). ImageNow 6.6.x and higher enable the use of a custom property at capture time. The connector references this value to set the related AP Invoice eForm XML data.

Example

```
<!--sourcetype = [index|customproperty] -->
<locationcode sourcetype="index" sourcename="f4"/>
```

eForm field mapping

The Worksheets or eForm settings are preconfigured to work with AP Invoice eForm. If you work with a custom eForm, you may need to adjust the settings to satisfy your eForm XML requirements.

The settings are organized in three sections: Header, PO lines, and GL Lines. The GL Lines section includes an example mapping. For each line, configure the following attributes.

- **node**. The name of the AP invoice eForm XML value.
- **datatype**. Use `"STRING"` for alphanumeric or date values or `"NUMBER"` for numeric values. The default setting is "STRING".
- **default**. The default value if the corresponding value from Intelligent Capture is null.
- **function.** This allows you to include a Java function such as **to lowercase** or **round**.
- **mapping**. used to convert values from Intelligent Capture.

Example

```
<header>
       <field xpath="/BrainwareDocument/InvHeader/INVOICETYPE" node="InvoiceType"
       datatype="" default="" function="" mapping="{'PO': 'po', 'NO-PO':'nonpo'}"/>
       <field xpath="/BrainwareDocument/InvHeader/INVOICENO" node="InvoiceNumber"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/INVOICEDATE" node="InvoiceDate"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/PONUMBER" node="PONumber"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/VENDORID" node="VendorID"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/VENDORNAME" node="VendorName"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/COMPANYCODE" node="CompanyNumHdr"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/FREIGHT" node="FreightAmount"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/TAX" node="TaxAmount"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/MISCCHARGE" node="MiscCharge"
       datatype="NUMBER" default="" function="" mapping=""/>
       field xpath="/BrainwareDocument/InvHeader/AMOUNT" node="TotalAmount"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/CURRENCY" node="Currency"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvHeader/SITEID" node="RemitTo"
       datatype="STRING" default="" function="" mapping=""/>
</header>

<polines lineitemxpath="/page/POLines/POLine">
       <field xpath="/BrainwareDocument/InvLines/LINENO" node="PO_LineNumber"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvLines/LINENO" node="LineNumber"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvLines/DESCRIPTION" node="ItemDescription"
       datatype="STRING" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvLines/QUANTITY" node="Quantity"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvLines/UOM" node="UOM" datatype="STRING"
       default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvLines/UNITPRICE" node="UnitPrice"
       datatype="NUMBER" default="" function="" mapping=""/>
       <field xpath="/BrainwareDocument/InvLines/TOTAL" node="ExtendedAmount"
       datatype="NUMBER" default="" function="" mapping=""/>
</polines>

<gllines lineitemxpath="/page/POLines/POLine/GLLines/GLLine">
       <field xpath="/BrainwareDocument/GLLines/GLACCOUNT"
       node="/page/POLines/POLine/GLLines/GLLine/GL1" datatype="STRING" default=""
       function="" mapping=""/>
       <field xpath="/BrainwareDocument/GLLines/COSTCENTER"
       node="/page/POLines/POLine/GLLines/GLLine/GL2" datatype="STRING" default=""
       function="" mapping=""/>
       <field xpath="/BrainwareDocument/GLLines/TOTAL"
       node="/page/POLines/POLine/GLLines/GLLine/LineAmount" datatype="NUMBER"
       default="" function="" mapping=""/>
</gllines>
```

Custom mapping

The IC_ReindexDoc script loads index values and custom properties according to the configuration settings in the IC_Config.xml file. After indexing and setting custom properties, the IC_ReindexDoc script executes the IC_Custom_WS_Mapping script. The IC_Custom_WS_Mapping script allows you to script a user exit for custom logic.

To add custom logic, create a routine in the IC_Custom_WS_Mapping.js script to contain the logic. Then, configure in IC_Config.xml as a <functiontoapply> inside the worksheet node for the custom eForm.

Example

```
<functiontoapply name="doAPInvoiceMappingCustom">
       <param1>abcdefghijklmn</param1>
       <param2>opqrstuvwxyz</param2>
</functiontoapply>
```

Configure value added tax (VAT) features

AP Invoice eForm, version 10.4.x and higher supports value added tax (VAT) fields and validation. If you use the Perceptive Intelligent Capture for Invoices solution with AP Invoice eForm, you can enable the following features.

- Enable currency and VAT amount validation
- Map VAT registration numbers (VAT IDs) to the AP Invoice eForm header fields
- Map VAT rates and codes to the AP Invoice eForm line-item fields
- Display VAT line-item fields in the AP Invoice eForm

To enable any of these features, refer to the following sections.

Enable currency and VAT amount validation

To enable currency or VAT amount validation in Intelligent Capture for each doctype, such as invoice or credit memo, complete any of the following steps.

- To validate the captured invoice currency against the original PO, in **IC_Config.xml**, under `<validation>`, set `<currency>` to `TRUE`.

  Note If the captured currency fails validation, Intelligent Capture Connector transfers the ERR_CURR_MATCH error message, which then displays as a message in AP Invoice eForm.

To allow Intelligent Capture to validate the header-level VAT with line-level VAT, in **IC_Config.xml**, under `<validation>`, set `<vatsummarytotal>` to `TRUE`.

> **Note** If the line-level VAT Amounts fields fail validation against the header-level VAT Total field, the Intelligent Capture Connector transfers the ERR_VAT_TOTAL_MATCH error message, which then displays as a message in the AP Invoice eForm.

Map VAT header fields

To map the Intelligent Capture-extracted VAT registration numbers (VAT IDs) to the AP Invoice eForm header fields, complete the following steps.

- In **IC_Config.xml**, under the `<doctype>` element, under `<header>`, uncomment (if existing) or add the following `<field>` elements.

  ```
  <field xpath="/BrainwareDocument/InvHeader/VENDORVATREGNO" node="VendorVATID"
  datatype="STRING" default="" function="" mapping=""/>
  <field xpath="/BrainwareDocument/InvHeader/BILLTOVATREGNO" node="SoldToVATID"
  datatype="STRING" default="" function="" mapping=""/>
  ```

- In the **Intelligent Capture database**, in the **BRWEXPHeader** table, add the following rows.
  - For the Vendor VAT registration number, add a row with the **FieldName** `VendorVATRegNo` and the **XMLTag** `VENDORVATREGNO`.
  - For the Bill-to VAT registration number, add a row with the **FieldName** `BillToVatRegNo` and the **XMLTAG** `BILLTOVATREGNO`.
- In the Intelligent Capture project file **DFI.sdp**, on script tab **3**, in the sub-routine **UserExitXMLOutput**, under **Case cXMLInvHeader**, add the following lines.

  ```
  fnWriteXMLField("HCVendorVATRegNo", pWorkdoc.Fields("VendorVATRegNo").Text)
  fnWriteXMLField("HCBillToVATRegNo", pWorkdoc.Fields("BillToVATRegNo").Text)
  ```

Map VAT line-item fields

To map the Intelligent Capture-extracted VAT rates and codes to the AP Invoice eForm line-item fields, modify the IC_Config.xml file. Complete any of the following steps.

- To map the line items for PO lines, in **IC_Config.xml**, under the `<doctype>` element, under `<polines>`, uncomment the following `<field>` elements.

  ```
  <field xpath="/BrainwareDocument/InvLines/TAXJURCODE" node="POLVATJurisdiction"
  datatype="STRING" default="" function="" mapping=""/>
  <field xpath="/BrainwareDocument/InvLines/TAXCODE" node="POLVATCode"
  datatype="STRING" default="" function="" mapping=""/>
  <field xpath="/BrainwareDocument/InvLines/TAXRATE" node="POLVATRate"
  datatype="NUMBER" default="" function="" mapping=""/>
  ```

- To map the line items for GL lines, in **IC_Config.xml**, under the `<doctype>` element, under `<gllines>`, add the following `<field>` elements.

  ```
  <field xpath="/BrainwareDocument/GLLines/TAXJURCODE"
  node="/page/POLines/POLine/GLLines/GLLine/GLVATJurisdiction" datatype="STRING"
  default="" function="" mapping=""/>
  <field xpath="/BrainwareDocument/GLLines/TAXCODE"
  node="/page/POLines/POLine/GLLines/GLLine/GLVATCode" datatype="STRING" default=""
  function="" mapping=""/>
  ```

Display VAT line-item fields in the eForm

To display the Intelligent Capture-extracted VAT jurisdictions and codes in the AP Invoice eForm, modify the AP_Skin.xml file. Complete the following step.

Prerequisite The `<polineitemprocessing>` element must be set to true in the IC_Config.xml file.

- In **AP_Skin.xml**, under **<fields>**, for each of the following `<poline>` elements, set the `visible` attribute to `true`.

  ```
  <poline label="lblPO_VATJurisdiction" visible="true">
  <poline label="lblPO_VATCode" visible="true">
  ```