Perceptive Intelligent Capture for Invoices

Integration Guide

Compatible with the following versions:
ImageNow 6.5.1 to 6.7.x
Intelligent Capture 5.5 SP2
Intelligent Capture for Invoices 2.2.x and higher

perceptivesoftware

Table of Contents

Product Overview

Intelligent Capture for Invoices is Perceptive Software's advanced solution for invoice processing. This solution fully integrates ImageNow with Intelligent Capture. The ImageNow components used include ImageNow Workflow, iScripts, External Message Agent, and Output Agent. In addition, Intelligent Capture provides a pre-developed project for invoice processing. The Perceptive Intelligent Capture for Invoices solution includes Intelligent Capture's production rules-based capture logic and the ability to deliver document status messages to the External Message Agent table.

The workflow process for an invoice includes the following steps.

- The user captures an invoice into ImageNow using a capture profile. The profile captures the document into an initial workflow queue. For example, **Send to Intelligent Capture**.

- The **Send to Intelligent Capture** queue contains an iScript, which runs each time an invoice enters the queue. With Output Agent, the iScript exports the invoice as an image to an Intelligent Capture-monitored directory. After the export, ImageNow routes the original document to the workflow queue **Intelligent Capture Processing**.

- The configured Intelligent Capture service monitors and imports the invoice image from the directory. Intelligent Capture completes the Classification, Extraction, and Verification processes on the invoice image. Then it generates one of the following external messages reflecting the invoice image status.

  - **Errors**

  - **Needs Verification**

  - **Success**

- After receiving the external message, External Message Agent executes an iScript. The iScript routes the invoice, in workflow, to one of the corresponding queues.

  - **Intelligent Capture Success**

  - **Intelligent Capture Verify**

  - **Intelligent Capture Error**

- When an invoice enters the **Intelligent Capture Success** queue, the iScript populates AP Invoice eForm and sets document and custom properties with valid, extracted data. 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 tool. After the user verifies that Intelligent Capture correctly extracted the invoice data, Intelligent Capture sends a new external message. External Message 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 External Message Agent iScript populates AP Invoice 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. The routing rule can then send the document to a queue for the user who handles new vendors.

Below is an example of a primary workflow. Depending on your invoice processing needs, your process may vary.
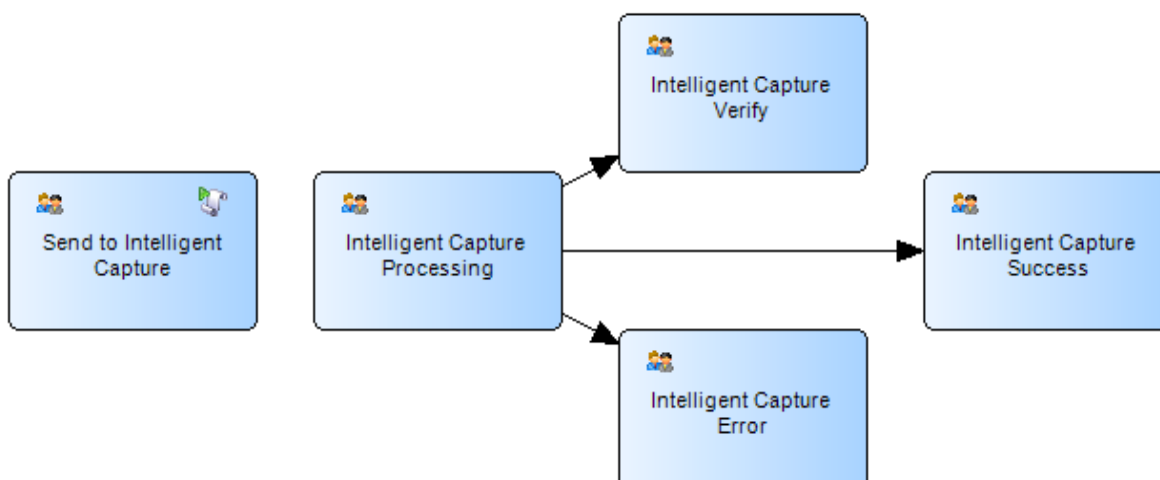
Prepare to install the solution

Solution prerequisites

Before downloading and installing the Intelligent Capture for Invoices solution, review and complete the following prerequisites, as necessary.

Program prerequisites

To install the Intelligent Capture for Invoices solution, you must install and have access to the following programs.

- ImageNow Server and Client 6.5.1. or higher

- Output Agent 6.5.1 or higher

- Intelligent Capture 5.5 SP2 or higher

- Intelligent Capture for Invoices 2.2 or higher

For information on the compatibility of specific versions of the ImageNow and Intelligent Capture, refer to the Compatibility matrix for the complete solution.

Task prerequisites

In ImageNow Management Console, create the Invoice and Credit Memo document types. For more information, refer to the "Add a document type" in ImageNow Help.

Determine component versions

To determine the compatibility of Intelligent Capture, Intelligent Capture for Invoices, and 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*:]\inserver6\<*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 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 2.x version

If you have previously installed Perceptive Intelligent Capture for Invoices version 2.x and want to determine your exact version number, refer to the Intelligent Capture for Invoices 2.2 Solution Guide, section 4.2.49, "PICI_VersionHistory."

Compatibility matrix for the complete solution

The following table maps the compatible versions of each component in the Intelligent Capture for Invoices solution.

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 first and then the AP Invoice eForm second. Keep AP Invoice eForm and Intelligent Capture Connector on the same build number.

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

ImageNow Server	AP Invoice eForm	Intelligent Capture Connector	Intelligent Capture	Intelligent Capture for Invoices
6.5.1.149 to 6.6.x	10.0.0.28 to 11.1.1.248	10.0.0.28 to 11.1.1.248	5.5 SP2 or higher	2.2
6.7.x	10.1.0.307 to 11.1.1.248	10.1.0.307 to 11.1.1.248		

Download the Perceptive Intelligent Capture files

To download the Intelligent Capture files, complete the following steps.

1. Go to the Perceptive Software website at www.perceptivesoftware.com, and log in to the **Customer Portal**.

2. In the **Product Downloads** page, search for and download the following files to a temporary directory. To determine which version of each file you need, refer to the Compatibility matrix for the complete solution.

   - Intelligent Capture

   - Intelligent Capture for Invoices

   - Intelligent Capture Connector

   - *Perceptive Intelligent Capture for Invoices Release Notes*

   - *AP Invoice Processing Solution Readme*

Rename the Intelligent Capture license files

You must rename the Intelligent Capture license files to work with the connector. Complete the following steps.

1. Navigate to **[*drive*:]\Program Files\Perceptive\Components\Cairo**.

2. Rename **zCroDemo.lic** to **zCroDemo.bak**.

3. Paste the **Perceptive Software Inc*.lic** file into the directory.

4. To register the Cairo (Cro*.dll), Cedar (Cdr*.dll), and Web Verifier (Bwe*.dll) components, complete the following substeps.

   - In **[*drive*:] \Program Files\Perceptive\Components\Cairo**, execute **RegCro.bat**.

   - In **[*drive*:]\Program Files\Perceptive\Components\Cedar**, execute **RegCdr.bat**.

   - In **[*drive*:]\Program Files\Perceptive\Components\Bwe**, execute **BweReg.bat**.

Verify Perceptive Intelligent Capture services

Verify that the Runtime Service Manager and System Monitoring services are both running correctly. For complete steps on verifying these systems, refer to the Perceptive Intelligent Capture Runtime Server User Guide, section 3.2, "Configuring the Runtime Service Manager."

Configure Perceptive Intelligent Capture for Invoices

To configure Intelligent Capture for Invoices for the connector, complete the following sections.

1. Configure the initialization file
2. Configure the Intelligent Capture database
3. Configure the ASSA column mappings and create ASSA pools

Configure the initialization file

To configure the Intelligent Capture initialization file, complete the following steps.

1. Navigate to your Perceptive Intelligent Capture for Invoices install directory. From the **Global** directory, open **DFI.ini**.

2. To configure the ASA settings, complete the following substeps.

   Note For more information on ASSA pools, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, section 11.3.6, "Appendix E - Configuring the Vendor ID Field without using a partition."

   - Verify **ASA_OP_01_AlphaNum** and **ASA_OP_03_AlphaNum** are set to `YES`.

   - Configure **ASA_VL_01_POOLPATH** and **ASA_OP_03_POOLPATH** to point to the pool directory you created during the Perceptive Intelligent Capture for Invoices installation. For example, set the path to `C:\Program Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\Pool`.

   - Configure **ASA_ VL_01_ImportPathFilename** to point to the **vendor.csv** file in the installed directory. For example, set the path to `C:\Program Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\vendor.csv`. You create this CSV file later with the Configure Virtual Table data sharing (optional) section.

     For example:

     ```
     # ASSA Settings for vendor and employee look-up

     ASA_VL_01_Class=Invoices
     ASA_VL_01_Fieldname=VendorASSA
     ASA_OP_01_AlphaNum=Yes
     ASA_OP_01_PoolRelative=YES
     ASA_VL_01_PoolPath= C:\Program
     Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\Pool
     ASA_VL_01_PoolDirectory=Pool
     ASA_VL_01_PoolName=Vendor
     ASA_OP_01_FileRelative=YES
     ASA_VL_01_ImportPathFilename= C:\Program
     Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\vendor.csv
     ```

   - Configure **ASA_OP_03_ImportPathFilename** to point to the **companycode.csv** file in the installed directory. For example, set the path to `C:\Program Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\companycode.csv`. You create this CSV file later in this guide in the Configure Virtual Table data sharing (optional) section.

3. To configure the document import (IMP_VL) settings, complete the following steps.

Note You configure the values of these components later in this guide with the Batch separation section.

- Set **IMP_VL_URN** to `COMPONENT1`.
- Set **IMP_VL_ClientID** to `COMPONENT2`.
- Set **IMP_VL_ExternalBatchID** to `COMPONENT3`.
- Set **IMP_VL_TransactionID** to `COMPONENT4`.
- Set **IMP_VL_TransactionType** to `COMPONENT5`.
- Set **IMP_VL_Separator** to `^`.

4. To configure the Database Connection Groups, complete the following steps.

- Set **SQL_VL_01_ConnectionString** to the Perceptive Intelligent Capture database. Example:

```
SQL_VL_01_ConnectionString=Provider=SQLOLEDB.1;Password=imagenow;Persist
Security Info=True;User ID=sa;Initial Catalog=Perceptive_Database;Data
Source=localhost
```

- Set **SQL_VL_02_ConnectionString** to the ImageNow database. Example:

```
SQL_VL_02_ConnectionString=Provider=SQLOLEDB.1;Password=imagenow;Persist
Security Info=True;User ID=inuser;Initial Catalog=INOW6;Data Source=localhost
```

5. To set the reporting key, set **REP_VL_ReportingKey** to `URN`.

6. To configure the Perceptive Content Integration settings, complete the following steps.

- Set **PCO_OP_WriteExternalMessages** to `YES`.
- Set **PCO_VL_SQLConnectionGroup** to the ImageNow database Connection String group. In this example, set it to `02`.

7. Save and close **DFI.ini**.

Configure the Intelligent Capture database

To configure the Intelligent Capture database for the solution, open the database and complete the following steps.

1. In the **BRWDAT table**, complete the following steps.

- Configure the **OutputDateFormat** column with `MMDDYYYY`.
- Configure the **OutputDateSeparator** with `/`.

2. In the **BRWBTO table**, verify that you have configured the **ProfileID** and **Distance** columns. For more information, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, section 4.2.4, "BRWBTO."

3. In the **BRWBTOFormats** table, verify that you have configured the **ProfileID**, **IndexID** and **Format** columns. For more information, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, section 4.2.5, "BRWBTOFormats."

4. In the **BRWSRC** table, verify that you configured the columns necessary for vendor searches. For more information, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, section 4.2.17, "BRWSRC."

5. In the **BRWVND** table, complete the following steps.

- Set **ValidateFromASSA** to `YES`.
- Set **CheckConditionVendors** to `NO`.
- Set **AlphaNumSiteSeparator** to `-`.
- Set **IgnorePOVendor** to `NO`.

6. In the **BRWPONFormats** table, verify that you have set your PO number formats. For more information, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, section 4.2.2, "BRWPONFormats."

7. In the **BRWLPR** table, complete the following steps.

- Set **DoLinePairing** to `YES`.
- Set **DoLinePairingForService** to `NO`.
- Set **GETPOLinesfromSAP** to `NO`.
- Set **CheckForMultiplePOs** to `NO`.
- Set **DescriptionThreshold** to `30`.
- Set **DescriptionDistance** to `10`.

8. In the **BRWEXP** table, complete the following steps.

- Set **XMLFilename** to `URN`.
- Set **XMLFileType** to `XML`.
- Set **XMLEncodingHeader** to `<?xml version="1.0" encoding="UTF-8"?>`.
- Set **XMLFileHeader** to `BrainwareDocument`.
- Set **XMLInvoiceHeader** to `InvHeader`.
- Set **XMLLineItemsHeader** to `InvLines`.
- Set **XMLLineItemsTag** to `Line`.
- Set **XMLGLLinesHeader** to `GLLines`.
- Set **XMLGLLinesTag** to `GLLine`.
- Set **XMLTaxHeader** to `TaxLines`.
- Set **XMLTaxTag** to `TaxLine`.

Note For information about the data exported in the XML, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, sections 4.2.31 to 4.2.34. For information about configuring XML exports, refer to section 7, "Configuring Data Export."

Configure the ASSA column mappings and create ASSA pools

To configure the ASSA column mappings and create ASSA pools for the connector, complete the following two sections. For more information on configuring the Vendor and Company Code ASSA pools, refer to the Perceptive Intelligent Capture for Invoices Solution Guide 2.2, section 11.

VendorASSA

To configure the designer to work with the Vendor Remit To Addressing format, complete the following steps.

1. In **Intelligent Capture Designer**, click the **Definition Mode** button, click **Invoices**, and then click the **Fields** tab.

2. Right-click the **VendorASSA** field and select **Show Properties**.

3. In the **Properties** window, click the **Analysis** tab.

4. On the **General** tab, select the **Enable alphanumeric identifier** check box.

5. On the **Import** tab, select the **Import field names from the first row** check box.

6. In the **CSV file** box, click the **[…]** button, navigate to the **Global** folder, and click **vendorlabels.csv** > **Open** > **Import**.

7. On the **General** tab, verify that the column headings from the CSV file display correctly in **Column name**.

8. In the **VENDORID** row, deselect the **ID** option.

   Note If using a custom eForm product, your VendorID column name may differ.

9. In the **IDENTIFIER** row, select the **ID** option.

   Note The **IDENTIFIER** row contains the new vendor remit-to key value and is the new search ID for vendor lookups.

10. In the **Class name format** box, verify that the text appears in the following format, even if the values differ:

    `[VENDORNAME]_[VENDORID]`

11. On the **Import** tab, click **Import** and browse to the **vendor.csv** file.

    Note If there are no headings in the CSV file, deselect the **Select Import field names from first row** option.

CompanyCode

To configure the designer to work with the Company Code or Business Unit format, complete the following steps.

1. In **Intelligent Capture Designer**, click the **Definition Mode** button, click **Invoices**, and then click the **Fields** tab.

2. Right-click the **CompanyCode** field and select **Show Properties**.

3. In the **Properties** dialog box, click the **Analysis** tab.

4. On the **General** tab, select the **Enable alphanumeric identifier** check box.

5. On the **Import** tab, select the **Import field names from the first row** check box.

6. If you did not create the initial CSV file, go to the CSV file section, navigate to the **Global** folder, select **companycodelabels.csv**, and then click **Import**.

7. Click the **General** tab and verify that the column headings from the CSV file display correctly in **Column name**.

8. In the **CompanyCode** or **BusinessUnit** row, select the **ID** option.

9. In the **BillToName** row, select the **Search** option.

10. On the **Import** tab, click **Import** and browse to the **companycode.csv** file.

    Note If there are no headings in the real CSV file, clear **Select Import field names from first row**.

Configure Perceptive Intelligent Capture

To configure Intelligent Capture, complete the following tasks.

- **Configure the run-time services**
- **Create a Verifier instance (DVS file)**

Configure the run-time services

To create and configure the run-time services, complete the following steps.

1. Start the **Perceptive Intelligent Capture Runtime Service**.

2. Right-click **Runtime Server Administration** > **New RTS Group**.

3. Name the new RTS group `Intelligent Capture`.

4. Right-click the **Intelligent Capture** group and select **New Machine**.

5. In the **Group Management** window, under **Available Perceptive Intelligent Capture**, type `localhost` or the server IP address and click **OK**. You should now see your computer name as an entry under the **Intelligent Capture** group. For example, SERVER01.

6. Right-click the new machine entry and click **New** > **RTS Instance**.

7. Name the new RTS instance `IMPORT_EXPORT`.

8. Right-click the new **RTS instance** and select **Properties**.

9. In the **Intelligent Capture Runtime Service Properties** dialog box, click the **General** tab and complete the following steps.

    - Under **Project File**, set **Use project file** to
      [*drive*:]\<*path*>\PerceptiveInvoiceSolution\Global.

    - Under **Directories**, select **Use database**, and from the **Database** list, select **JOB1**.

    - Under **Batch Scanning Delay and Mode**, review the **Wait** settings to determine how frequently Intelligent Capture checks for incoming invoices. We recommend setting this value to **5**.

    - Under **Automatic start / stop**, review the **Start at** and **Terminate at** times.

    - Under **Automated System Updates**, review the **Automatic pool update every setting**. This setting controls how frequently Intelligent Capture updates the Vendor table. Enable it only for the **ASSA_UPDATER** instance.

Note These optional settings can assist with automation. For more information, refer to the Perceptive Intelligent Capture Runtime Server User Guide.

10. On the **Workflow** tab, verify the following **Process Step** options.

   • For the **IMPORT_EXPORT** instance, select **Import** and **Export**.

   • For the **CLEANUP_800** instance, select **Cleanup**.

   • For the **ASSA_UPDATER** instance, do not select any options.

   • For the **OCR_CLASS_EXTR** instance, select **OCR**, **Classification**, and **Extraction**.

11. On the **Import** tab, complete the following steps.

   • Set **Import Directory** to `[`*`drive:`*`]\<`*`path`*`>\PerceptiveInvoiceSolution\Import`.

   • Under Document Grouping, select 1 folder per batch (no subdirectories).

   • Under Import Condition, select Always import documents.

   • Under **Extended Settings**, select **Limit batch size**, and set the **documents** value to `1`. This setting ensures that Intelligent Capture processes each invoice independently as its own batch. ImageNow holds batches of more than one invoice if at least one invoice in the batch has a verification or error issue. Each error-free invoice can flow through the process without blocking the remaining invoices.

12. To verify that the **Export** and **Cleanup** settings for the RTS are correct, complete the following substeps.

   1. Right-click the runtime service and select **Properties**.

   2. On the **OCR+Export+Cleanup** tab, under **Export**, clear **General Protocol File**.

13. You should create at least four RTS instances. After you create and configure the IMPORT_EXPORT instance, repeat the configuration steps to create the following instances.

   • **CLEANUP_800**

   • **ASSA_UPDATER**

   • **OCR_CLASS_EXTR**

   Note You can add additional RTS instances dedicated to OCR_Class_EXTR to accommodate large page volumes.

14. To save your changes, click **OK**.

Create a Verifier instance (DVS file)

To create a new Verifier instance, complete the following steps.

1. Open **Intelligent Capture Verifier**.

2. On the **File** menu, click **Save As**, then name the file **Verifier** and save it in the [*drive*:]\<*path*>**PerceptiveInvoiceSolution** folder.

3. In the toolbar, click **Options** and select **Settings**. In the **Intelligent Capture Verifier Properties** dialog box, complete the following substeps.

4. On the **General** tab, complete the following substeps.

    1. Under **Project File**, verify that the path associated with the project file refers to the loaded Intelligent Capture project. Example: **C:\Program Files\Perceptive\Projects\PerceptiveInvoiceSolution\Global\BW Invoices.sdp**.

    2. Under **Directories**, enable **Use database as documents and statistics source** and select **JOB1** from the list.

    3. On the **Workflow** tab, select **Classification verification** and **Extraction verification.**

    4. Click **OK**.

5. Exit the **Verifier** and click **Yes** to save your changes.

Configure the Verifier instance

To configure the Web Verifier, follow the instructions contained in the Intelligent Capture Web Verifier User Guide. Then, complete the Update the connection strings task below.

Update the connection strings

If you manually created your Oracle or MS SQL databases, use one of the following procedures to update the connection strings.

Update the Oracle database connections

To update the Oracle database connections, complete the following steps.

1. Navigate to the [*drive*:]**WINDOWS\Microsoft.NET\Framework\v2.0.507.27\CONFIG** directory and open **machine.config** with a text editor.

2. Under **<DbProviderFactories>**, add the following section.

   ```
   <add name="EF Oracle Data Provider invariant="EFOracleProvider" description="EF
   Provider for Oracle testing" type="EFOracleProvider.EFOracleProviderFactory,
   EFOracleProvider, Version=1.0.0.0, Culture=neutral,
   PublicKeyToken=def642f226e0e59b"/>
   ```

3. Save and close the **machine.config** file.

4. Navigate to [*drive*:]\<*path*>**Perceptive\Perceptive Intelligent Capture\bin**.

5. Open each of the following configuration files with a text editor.

- DstDsr.exe.config

- DstVer.exe.config

- DstSlm.exe.config

- DstHost.exe.config

- Brainware.System.Project.exe.config

6. In each configuration file, modify the following connection strings as needed to match the following code.

```
<connectionStrings>
<add name="Entities"
connectionString="metadata=res://*/Entity.ORAEntities.csdl|res://*/Entity.ORAEntiti
es.ssdl|res://*/Entity.ORAEntities.msl; provider=EFOracleProvider; Provider
Connection String='Data Source=percep; User ID=inuser; Password=imagenow'"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

7. Save and close each configuration file.

Update the MS SQL database connections

To update the MS SQL database connections, complete the following steps.

1. Navigate to [*drive*:]\<*path*>\Perceptive\Perceptive Intelligent Capture\bin\ directory.

2. Open each of the following configuration files with a text editor.

- DstDsr.exe.config

- DstVer.exe.config

- DstSlm.exe.config

- DstHost.exe.config

- Brainware.System.Project.exe.config

3. In each configuration file, modify the following connection strings as needed to match the following code.

```
<connectionStrings>
<add name="Entities"
connectionString="metadata=res://*/Entity.Entities.csdl|res://*/Entity.Entities.ssd
l|res://*/Entity.Entities.msl;provider=System.Data.SqlClient;provider connection
string=&quot;Data Source=<Database Server>;Initial
Catalog=Perceptive_Software_Database;User Id=<User
ID>;Password=<Password>;Integrated
Security=false;MultipleActiveResultSets=True&quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

4. Save and close each configuration file.

Install Perceptive Intelligent Capture Connector

Intelligent Capture Connector creates the link between ImageNow and Intelligent Capture for Invoices.

To install Intelligent Capture Connector, complete the following tasks.

1. Copy and load the connector scripts
2. Configure BW_ReindexDoc.js
3. Copy the XML files

Extract and load the connector scripts

To install the connector, extract and load the connector iScripts.

Important If upgrading Intelligent Capture Connector in an existing implementation, back up the existing files in the **[*drive*:]\inserver6\etc**, **\script**, and **\form** or **\worksheet** directories before installing the latest connector scripts.

Complete the following steps.

1. Navigate to the **Perceptive Intelligent Capture Connector.zip** file you previously downloaded.
2. Extract the **\inserver6\script** folder to the **[*drive*:]\inserver6\script** directory.

   Note If you installed AP Invoice eForm in an active-active server environment, extract to the **[*shared path*]\inserver6\script** directory.

   This folder loads the following iScripts.

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

     Note If upgrading the connector, replace this file with your backup copy that contains your custom scripting.

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

Configure BW_ReindexDoc.js

To configure the BW_ReindexDoc.js script to read the correct external message (EM) table properties, complete the following steps.

1. Open **BW_ReindexDoc.js** with a text editor.
2. Set **#define PROP_STATUS_CODE** to `"ic_status"`. Example:

```
#define PROP_STATUS_CODE "ic_status"
```

Copy the XML files

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

- Copy the contents of the **Perceptive Intelligent Capture Connector.zip\inserver6\etc\ap** folder and place it in the **[*drive*:]\inserver6\etc\AP** folder.

  Note If you installed AP Invoice eForm in an active-active server environment, extract to the **[*shared path*]\inserver6\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. For more information about configuring this file, refer to the Configure Virtual Table data sharing (optional) section.

- **BW_Config.xml** controls the basic functions of Intelligent Capture Connector.

Configure Perceptive Intelligent Capture Connector

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

1. Configure Output Agent
2. Configure External Messaging Agent
3. Configure ImageNow workflow
4. Configure core connector features: BW_Config.xml
5. 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 Output Agent

To configure Output Agent, complete the following steps.

1. Navigate to **[*drive*:]\inserver6\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*]\inserver6\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:***]\inserver6\etc** directory and open **inserverEM.ini** with a text editor.

   Note If working in an active-active server environment, navigate to **[***shared path***]\inserver6\etc**.

2. Insert the following lines into the INI file.

   ```
   [intelligentcapture]
   ic_status_update=BW_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 ImageNow.

Configure ImageNow 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 ImageNow requires, 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 **BW_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 BW_LI_Review.

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

```
If AP Detail Valid (CP) starts with "BW_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 core connector features: BW_Config.xml

The BW_Config.xml file controls the basic functions of Intelligent Capture Connector and the BW_ReindexDoc.js and BW_ExportDoc.js scripts.

If upgrading the connector, use your backup copy of BW_Config.xml to configure the latest version of this file with your values.

The following sections describe the configuration options.

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 BW_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 BW_ReindexDoc.js script stores the XML as a DCExport subobject of the ImageNow 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 BW_ReindexDoc.log file, set the debug level in BW_ReindexDoc.js to 5 (DEBUG).

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

Each ImageNow 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) features.

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

Batch separation

You can configure the BW_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 following format {Custom Property}.

Duplicate invoice detection: legacy invoice check

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

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

1. In **BW_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 ImageNow 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>
```

Duplicate invoice detection: advanced duplicate check

The advanced invoice check, for use with AP Invoice eForm, 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 BW_Config.xml with the advanced duplicate invoice check, complete the following steps.

1. In **BW_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 ImageNow 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
   ```

Example:

In this 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.

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

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 ImageNow 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:\\inserver6\\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 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 Virtual Tables Configuration Guide.

Location code

Use this setting to define the source of the Location code, which is provided at scan time. Location code is not provided by Intelligent Capture. 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 ImageNow 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 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 lets you 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 BW_ReindexDoc script loads index values and custom properties according to the configuration settings in the BW_Config.xml file. After indexing and setting custom properties, the BW_ReindexDoc script executes the BW_Custom_WS_Mapping script. The BW_Custom_WS_Mapping script allows you to script a user exit for custom logic.

For adding custom logic, create a routine in the BW_Custom_WS_Mapping.js script to contain the logic. Then, configure in BW_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 **BW_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 **BW_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 **BW_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 BW_Config.xml file. Complete any of the following steps.

- To map the line items for PO lines, in **BW_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 **BW_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 BW_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">
```

Configure Virtual Table data sharing (optional)

If your solution uses AP Invoice eForm, you can configure virtual table data sharing. To use the verification and search processes in Intelligent Capture, you load the Company and Vendor CSV files. These files automatically load using the virtual table data associated with the Perceptive Invoice eForm product.

After you run the AP_VirtualTable_Updater.js script to load the AP Invoice eForm virtual tables, the script places a formatted copy of both the vendor and company code files in the global directory. Update the AP_VirtualTable_Updater.xml file, as shown below, to create the Intelligent Capture data files.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration host="lawson" bwexportpath="C:\\Program
Files\\Perceptive\\Projects\\PerceptiveInvoiceSolution\\Global\\"
bwexportdelimiter=";" inputpath="C:\inserver6\temp\ap\import\*.csv"
archivepath="C:\inserver6\temp\ap\archive" errorpath="C:\inserver6\temp\ap\error">
  <recordstructure recordtype="BUS_UNIT" bwfile="busunit.csv"
table="Z_APW_APBusinessUnit">
    <column required="true" type="flag" customprop="Z_APW Active"/>
    <column enabled="true" customprop="Z_APW Business Unit ID" bwcolumn="1" />
    <column enabled="true" customprop="Z_APW Business Unit Name" bwcolumn="2,4" />
    <column enabled="true" customprop="Z_APW Vendor Group" bwcolumn="3" />
    <column enabled="true" customprop="Z_APW VAT Jurisdiction" bwcolumn="5" />
    <column enabled="true" customprop="Z_APW VAT ID" bwcolumn="6" />
  </recordstructure>
</configuration>
```

Company Code

Extracted from the Business Unit virtual table, the Company Code CSV table lists valid company codes.

To find a matching Business Unit, such as the "Bill To Name" on an invoice, and populate the corresponding company code, configure the CompanyCode ASSA search. The table must contain the following fields.

- Business Unit

- Company Code

For example:

	A	B
1	10;New Hanover Regional Med Ctr	
2	20;Carolina Healthcare Associates	
3	85;Coastal Carolina Heart	
4	200;Edwards Cancer Ctr	
5	400;Brainware	
6	500;Texas	

Vendor

The Vendor CSV table lists valid vendors extracted from the Vendor virtual table. The table contains the following fields.

- Vendor ID

- Remit To #

- Vendor Name

- Address Line 1

- Address Line 2

- City

- Postal Code

- State

- Country

- Identifier (*hyphenated combo field Vendor ID-Remit To#)

  Important The identifier field must be unique for each row in the Intelligent Capture Vendor CSV table.
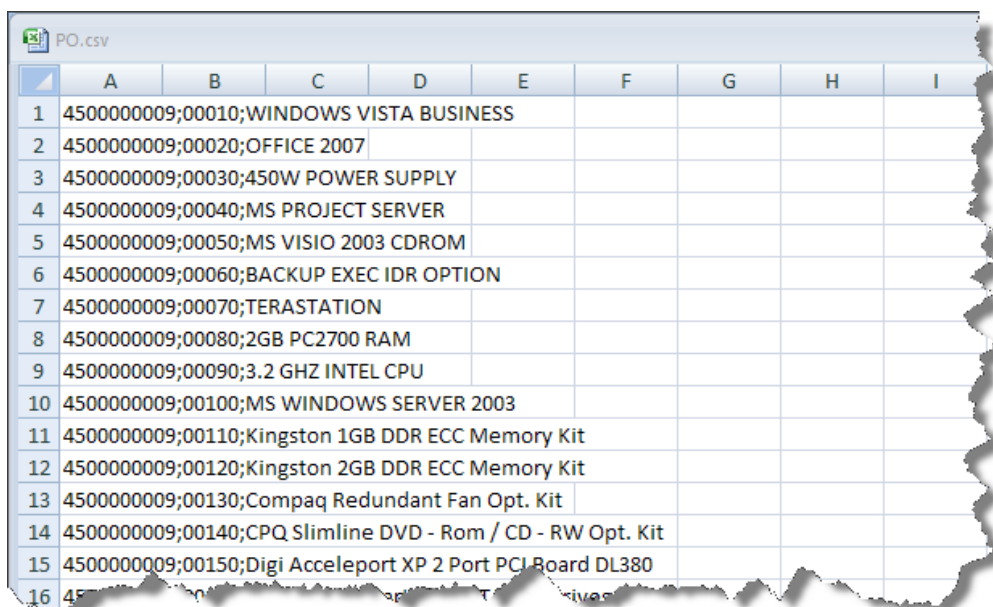
For example:

	A	B	C	D	E	F	G	H	I	J	K
1	10003;01;Acme;5 Dedrick Place;;West Caldwell;NJ;07006;;10003-01										
2	10005;01;Admiral Linen & Uniform Service;2030 Kipling;;Houston;TX;77098;;10005-01										
3	10009;01;ANGIODYNAMICS;P.O. Box 1549;;Albany;NY;12201-1549;;10009-01										
4	10010;60425-1586;Landauer;Inc.;2 Science Road;;Glenwood;IL;01;10010-60425-1586										
5	10011;01;Cook Medical Incorporated;22988 Network Place;;Chicago;IL;60673-1229;;10011-01										
6	10012;01;Boston Scientific;75 Remittance Drive;Suite 6094;Chicago;IL;60675-6094;;10012-01										
7	10015;01;Software House International;P.O. Box 8500-41155;;Philadelphia;PA;19178;;10015-01										
8	10016;01;Teksystems;P.O. Box 198568;;Atlanta;GA;30384-8568;;10016-01										
9	10018;01;Blueline Simulations LLC;Wrong Address;Wrong Address2;Wrong City;Wrong State;Wrong Zip;;10018-01										
10	10018;02;Blueline Simulations LLC;12400-2 Wake Union Church Road;Suite 32;Wake Forest;NC;27587;;10018-02										
11	10019;01;McManus Instrument Co. Inc;P.O. Box 829;;Village Mills;TX;77663;;10019-01										
12	10020;01;Promega;P.O. Box 689768;;Milwaukee;WI;53268-9768;;10020-01										
13	10004;01;Ricoh;5 Dedrick Place;;West Caldwell;NJ;07006;;10004-01										
14											
15											
16											
17											

PO Lines

The PO (Lines) CSV table lists valid PO lines extracted from the PO Lines virtual table. The table contains the following fields.

- Purchase Order #

- PO Line #

- PO Line Description

For example:

Configure Verifier batch security (optional)

To configure batch security, complete the follow sections.

Enable new columns for batch view

Four columns, ExternalGroupID, ExternalBatchID, TransactionID, and TransactionType, known as batch level entities, are available to hold additional information on batches.

By default, these columns are not shown. To display the columns in the batch view, execute the following commands in the Intelligent Capture database.

Syntax:

```
exec sp_SetGlobalApplicationSetting 'ColumnSettingName','Column Name to Display',
Enabled boolean
```

ExternalGroupID:

```
exec sp_SetGlobalApplicationSetting
'SysAppBatchColumnExternalGroupId','ExtGroupID',True
```

ExternalBatchID:

```
exec sp_SetGlobalApplicationSetting
'SysAppBatchColumnExternalBatchId','ExtBatchID',True
```

TransactionID:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnTransactionId','TransID',True
```

TransactionType:

```
exec sp_SetGlobalApplicationSetting
'SysAppBatchColumnExternalTransactionType,'TransType',True
```

Notes

- You can customize the ExtGroupID, ExtBatchID, TransID, and TransType values to meet your needs.

- To hide the Batch entities from the batch view, set the last value to False, and then run the same commands.

Update user group security

To set up users, refer to the Perceptive Intelligent Capture Solution Guide 2.2, section 14, "Setting up users."

Enable Windows and Database authentication

You can import local workstation and network Windows users into the Intelligent Capture project authentication sub-system. This allows automatic project authentication with the currently logged-in user account for all Intelligent Capture applications. You must enable Windows Authentication and Database Authentication to implement this batch security feature.

To enable Windows and Database authentication in Intelligent Capture, complete the following steps.

1. In the Intelligent Capture Designer application, open the project file.

2. From the menu, select **Options** > **Users, Groups and Roles**.

3. In the **Project Authentication Settings** dialog box, on the **Users** tab, select the **Allow Windows Authentication** and **Allow Database Authentication** check boxes.

4. Click **OK**.

Resolve Visibility port conflict

Visibility is the Intelligent Capture reporting and dashboard product. It includes several predefined reports, processing metrics and analysis. Visibility uses an embedded copy of Apache Tomcat. Tomcat uses the same ports that ImageNow often utilizes. To resolve any port conflicts, complete the following steps.

1. Navigate to the **C:\Program Files\Brainware Visibility\Jakarta\conf** folder and open the **server.xml** file in a text editor.

2. In the **<Connector>** element, set the **port** to `"8010"`.

    Example:

    ```
    <Connector port="8010" protocol="AJP/1.3" redirectPort="8444"/>
    ```