Perceptive Capture Connector

Installation and Setup Guide

Version: 2.0.x

Hyland™

Table of Contents

About the connector

Perceptive Capture Connector (PC Connector) allows you to create, configure, and enable channels that map data and functionality between Perceptive Content and Perceptive Intelligent Capture (PIC) for Invoices.

With a combination of channels, you can create a solution that enables exchange of files between Perceptive Content and PIC, routes documents in Perceptive Content according to their status code, and imports the extracted XML data into Perceptive Content eForm fields, document properties, or other objects.

Each PC Connector channel consists of a trigger, an action, and a results phase.

A trigger is an event that initiates the channel. The PC Connector provides two triggers.

- Document Trigger. This trigger allows you to create, export, and soft delete documents in Content and create channels that pass XML data back to your original documents in Perceptive Content.
- Status Update Trigger. This trigger allows you to create channels that trigger when PIC provides status codes during document processing.

An action is a task that the channel performs when the trigger event occurs. The PC Connector provides the following actions.

- Capture Document Action. This action allows you to create a document in Perceptive Content using the configured Capture Profiles and Application Plans.
- File System Document Export. This action exports any Perceptive Content documents from an Integration ASQ to a specific network location. . For example, you can use this action to create the TIFF files that PIC uses for data extraction.

The results phase lets you specify what to do with the data that is available after the action completes. For example, you can output extracted XML values to Perceptive Content document keys, custom properties, eForm fields, and the DCExport subobject.

This guide outlines the installation and configuration procedures for the PC Connector. It also includes basic instructions for creating and configuring each channel type.

The PC Connector files include several sample XML files to help you understand the way you can configure your channel mappings.

For more information about other required Perceptive Connect Runtime components, refer to the *Perceptive Connect Runtime Installation Online Help*.

Overview of the setup process

You can install the connector and set up the channels for enabling the connector to function for PIC.

To install the connector and set up File Output, Status Update, and Export channels, review or complete the following sections as needed.

- Install the connector
- Create PIC File Output channels
- Create PIC Status Update channels
- Create PIC Export channels

Install the connector

You can install Perceptive Capture Connector, version 2.0.2 and related files manually using Perceptive Connect Runtime Web Console or from the "Upload new bundles" section in Perceptive Connect Runtime. You must install the connector on the same computer as the Perceptive Connect Runtime service.

Obtain and store the installation files

You can obtain the Perceptive Capture Connector installation files and store them in a temporary directory on your computer for installation purposes.

To receive the installation files, complete the following step.

1. To obtain the installation files, contact the Hyland Software Technical Support group. For a list of Technical Support contact numbers, go to hyland.com/pswtscontact.

Install the connector using the drag-and-drop feature

To install the connector using Perceptive Connect Runtime, complete the following steps.

1. In a browser, type the address in the format `http://<Perceptive Connect Runtime machine name>:<port>`.
2. In the left pane, under **Manage**, click **Install a Connector**.
3. In **Perceptive Connect Runtime**, on the **Upload new bundles** page, drag the **Perceptive Capture Connector 2.0.2-all.zip** file to the area designated by the box. You can also click **Manual upload** to upload the files manually.

   Result Complete, queued, and processing files appear at the bottom of the page. After Perceptive Connect Runtime processes an item, a summary of the processed files, as well as an overview describing how many channels were affected by the installation, is displayed.

   > **Note:** The affected channel summary contains the following lists.

   - **Fixed**. Channels that previously failed to validate and became valid after the installation.
   - **Broken**. Channels that previously validated and became invalid after the installation.
   - **Unknown**. Channels that PCR cannot verify as valid or invalid for the report. Usually, this is a byproduct of older channels with trigger parameters that are not stored in the database as plain text.

4. Perform one of the following actions.
   - Click **Accept** to accept the installation.
   - Click **Roll back** to undo the installation.

   > **Note:** You must accept or roll back the installation before PCR can process the next item. The bundle installation summary contains the following categories.

   - **Installed**. Bundles installed from the file or archive and uploaded by the user.
   - **Failed**. Bundles that failed to install. This includes a message with the reason for failure.
   - **Skipped**. Files that PCR cannot install, such as non-JAR files.

Install the connector manually

You can install the Perceptive Capture Connector from the Perceptive Connect Runtime Web Console page on Perceptive Connect Runtime. You can also install the connector by clicking "Manual upload" in the "Upload new bundles" page.

To install the PC connector, complete the following steps.

1. In a browser, navigate to the **Perceptive Connect Runtime Web Console** URL in the format `http://<Perceptive Connect Runtime machine name or IP>:<port>`.

2. In the browser dialog box, enter the Web Console user name and password.

    > **Note:** The default user name and password are `admin`. However, the administrator can change the defaults during the Perceptive Connect Runtime installation process.

3. In the **OSGI Bundles** page, click **Install/Update**.

4. In the **Upload/Install Bundles** dialog box, complete the following substeps.

    1. Select the **Start Bundle** check box.
    2. Clear the **Refresh Packages** check box.

    > **Note:** The **Start Level** remains at the default value.

5. To install the JAR files included in the connector ZIP file, complete the following substeps.

    1. Click **Choose Files** and navigate to the connector in your temporary directory.
    2. Select the **capture-connector.content.mapping-2.0.2** and **capture-connector.endpoint-2.0.2** JAR files, one at a time, and click **Open**.
    3. Click **Install or Update**.

Upgrade the connector

You can upgrade Perceptive Capture Connector, version 2.0.2 and related files manually using Perceptive Connect Runtime Web Console or from the "Upload new bundles" section in Perceptive Connect Runtime. You must delete all the existing bundles and channels before upgrading the connector from version 1.x to 2.0.2.

Upgrade the connector using the drag-and-drop feature

To upgrade the connector using the drag-and-drop feature in Perceptive Connect Runtime, complete the following steps.

1. In a browser, type the address in the format `http://<Perceptive Connect Runtime machine name>:<port>`.

2. In the left pane, under **Manage**, click **Install a Connector**.

3. In **Perceptive Connect Runtime**, on the **Upload new bundles** page, drag the **Perceptive Capture Connector 2.0.2-all.zip** file to the area designated by the box. You can also click **Manual upload** to upload the files manually.

    Result Complete, queued, and processing files are displayed at the bottom of the page. After

Perceptive Connect Runtime processes an item, a summary of the processed files, as well as an overview describing how many channels were affected by the installation is displayed.

> **Note:** The affected channel summary contains the following lists.

- **Fixed**. Channels that previously failed to validate and became valid after the installation.
- **Broken**. Channels that previously validated and became invalid after the installation.
- **Unknown**. Channels that PCR cannot verify as valid or invalid for the report. Usually, this is a byproduct of older channels with trigger parameters that are not stored in the database as plain text.

4. Perform one of the following actions.
   - Click **Accept** to accept the installation.
   - Click **Roll back** to undo the installation.

> **Note:** You must accept or roll back the installation before PCR can process the next item. The bundle installation summary contains the following categories.

- **Installed**. Bundles installed from the file or archive and uploaded by the user.
- **Failed**. Bundles that failed to install. This includes a message with the reason for the failure.
- **Skipped**. Files that PCR cannot install, such as non-JAR files.

Upgrade the connector manually

To upgrade from a previous version of Perceptive Capture Connector, complete the following steps. You can also upgrade the connector by clicking "Manual upload" in the "Upload new bundles" page.

1. Uninstall the existing version of the connector
2. Install the latest version of the connector

Uninstall the existing version of the connector

To uninstall the existing version of the connector, complete the following steps.

1. In the left pane, under **TROUBLESHOOT**, click **List OSGi Bundles**.

2. In the **OSGI Bundles** page, click **Stop** ▪ next to the following bundles.
   - **Perceptive Capture Connector Content Mapping**
   - **Perceptive Capture Connector Endpoint**
   - **Perceptive Capture Connector Writer**

3. Click **Uninstall** 🗑 next to each of the stopped bundles.

Install the latest version of the connector

To install the latest version of Perceptive Capture Connector, complete the following steps.

1. Locate the temporary directory you created in the "Download and extract the files" section.
2. In the left pane, under **TROUBLESHOOT**, click **List OSGi Bundles**.
3. In the **OSGI Bundles** page, click **Install/Update** and complete the following substeps.

1. In the **Upload/Install Bundles** dialog box, click **Choose Files**.
2. Navigate to the **capture-connector.content.mapping-2.0.2.jar** and **capture-connector.endpoint-2.0.2.jar** files in the extracted directory, click **Open**, select the **Start Bundle** check box, and then click **Install or Update**.
3. Restart Perceptive Connect Runtime service.

Configure the connector

To configure the Perceptive Capture Connector, complete the following steps.

1. In the **Connect Runtime Web Console**, in the left pane, under **Manage**, click **Configure**.
2. In the **Name** column, under **Perceptive Capture Connector**, click **Edit the configuration values** .
3. In the **Client Hostname** box, type the hostname or IP address of the system where PIC is installed.
4. In the **Channel Timeout** box, type the maximum time in seconds, for which the client will wait for the channels to finish execution.
5. Click **Save**.

Configure the connector for PIC

Perceptive Content workflow with Envoy services enables Perceptive Content to communicate with Perceptive Connect Runtime (PCR), and the file export directory. Perceptive Intelligent Capture collects the document from the file export directory. The Integration automated system queue you implement in your workflow process sends a web service notification using the Envoy service operation name you define. Based on the format of the document to be processed, PCR receives the document from Perceptive Content. After this process is complete, PCR sends a success response using Perceptive Integration Server. Perceptive Content can route the file forward in the Perceptive Content workflow for additional processing.

To configure Perceptive Content to use the connector, complete the following procedures.

1. Configure PIC for content integration
2. Configure Envoy Services in Perceptive Content
3. Configure the Content Connect Service
4. Create the workflow process
5. Create the queues

Configure PIC for content integration

You must configure PIC for content integration by updating the PIC database settings.

To update the database settings, complete the following step.

1. Set `BRWAMT.ExportThousandSeparator` with the value "" and `BRWAMT.ExportDecimalSeparator` with the value ".".

Configure Envoy Services in Perceptive Content

If you have Perceptive Content version 7.1.5 or earlier installed on your machine, you must configure the Perceptive Content Envoy web service—`AsqEndpointService with operation—InvokeTrigger and WSDL URL—http://<server IP:port>/ws/workflowTrigger?wsdl`.

To configure a Perceptive Content Envoy service, complete the following steps.

1. In **Perceptive Content Management Console**, in the left pane, click **Envoy Services** and then click **New**.

2. In the **Envoy Services** dialog box, in the **Definition** page, set the following attributes.

   1. In the **Name** box, type a name to identify the remote service. For example, `ASQEndpointService`.

   2. Optional. In the **Description** box, type a description.

   3. In the **URI** box, type the URI for your connector server in the format `http://<Perceptive Connect Runtime server>:<port>/ws/workflowTrigger?wsdl`.

   4. In the **Authentication** list, select **None**.

   5. Optional. If you want to enable interceptor logging for the remote service, select the **Enable interceptor logging** check box.

3. Click **Next**.

4. In the **Operations** page, in the **ASQEndpointService** operation, select the check box for **Invoke** and click **Finish**.

   > **Note:** In case of **ErpCommonOperationsService**, all its associated operations load in the **Operations** page.

5. Click **Finish**.

Configure the Content Connect Service

Perceptive Content version 7.1.5 and higher, includes a new queue type called Connect ASQ, which can be configured to integrate with a single instance of the Connect Runtime.

To configure the Perceptive Content Connector service, complete the following steps.

1. Navigate to the location where the Content Server is installed and enter the **{install location}\etc** directory.

2. Open the `inserverWorkflow.ini` file and make the following changes in configuration settings.

   - Configure the `connect.uri` setting and set it to your Connect Runtime instance with the format `http://{Connect Runtime host name}:{port}/rs/workflowTrigger`.

   - Configure the `connect.timeout` to your desired expiration time.

3. Save the file.

4. Optional. The setting is automatically loaded after a short period. To reload the configuration immediately, restart the Content Server service.

Create the workflow process

To create a workflow process, complete the following steps.

1. In **Perceptive Content Management Console**, in the left pane, click **Workflow**.
2. In the right pane, on the **Workflow** tab, click **New**.
3. In the **Add Process** dialog box, complete the following substeps.
    1. In the **Name** box, type a name.
    2. In the **Description** box, type the description of the process that appears in the tooltip for the process.
4. Click **OK**, select the process, and then click **Modify**. Use **Perceptive Content Workflow Designer** to create queues, routes, and workflow rules.

Create the queues

Based on how you want to process your invoices, create the integration ASQ, connect ASQ, and the work queues. If you have Perceptive Content 7.1.5 or earlier installed on your machine, you can create an integration ASQ for the submit queue and from version 7.1.5 onwards, you can create a connect ASQ for the submit queue. The workflow you create using the integration ASQ, connect ASQ, and work queues ascertains how a document is processed. The ASQ integrates the workflow process with Perceptive Connect Runtime and uses the subsequent work queues to indicate the processing status of the document.

For information about creating a workflow process and queues, refer to the following help topics.

- Create an Integration ASQ
- Create a Connect ASQ
- Create a work queue

Note: For a sample workflow, refer to the Example of Perceptive Content workflow for Perceptive Intelligent Capture section.

Create PIC File Output channels

File Output channel exports copies of documents from Perceptive Content to a local or network directory, triggered by an Integration ASQ. PIC is configured to monitor the directory and pull incoming TIFF files, and then performs classification, extraction, and validation on each image. The channels from version 1.2.5 are not compatible with version 2.0.2, however a part of the XML is identical.

This section provides the steps for creating and configuring a File Output channel by specifying the ASQ used for the trigger, the destination directory, the separator character, and the file name parameters. The file name always includes the Content document ID, and can include a separator character, such as ^. The file name can also include up to 99 value segments, as needed, using any of the following options.

- A literal string
- A Perceptive Content document property, which can include the document name and document version number
- A Perceptive Content document key value
- A Perceptive Content custom property value

You can create multiple File Output channels with differing value segments or destination directories as needed for your solution. Before you begin configuring a channel, we recommend that you open the *Perceptive Connect Runtime Installation Online Help* to use as reference material.

Channel requirements

A File Output channel requires the following components.

- An Integration ASQ ID from Perceptive Content
- The Integration ASQ trigger and the Document Property reader provided by the Perceptive Content Connector
- The File System Document Export action provided by this connector
- The Trigger reader provided by the Perceptive Connect Runtime

Create and configure a File Output channel

To create and configure a File Output channel, complete the following steps.

1. In a browser, navigate to the channel wizard URL in the format http://<Perceptive Connect Runtime machine>:<port>.
2. In the left pane, under **Manage**, click **Create a Channel**.
3. Optional. In the right pane, in the **Channel** section, enter a name and description for the channel.

   Note: Entering a name and description for the channel may help you find the channel on the **List Channels** page.

4. In the **Trigger Information** section, complete the following substeps.
   1. In the **Trigger** list, select **Integration ASQ Trigger**.
   2. To specify the queue that triggers the action, in the **Workflow Queue ID** box, type the ASQ ID, such as 301YW2R_0001LLH16000028, and click **Continue**.
5. On the **Modify Channel Mapping** page, under **Actions** , select **FileSystemDocumentExport**.
6. From the temporary directory where you downloaded the connector files, open the sample *Export_ Inputs.xml* file in the **FileSystemDocumentExport** folder with a text editor.
7. To configure the inputs, in the **Inputs** XML editor, use the sample XML file and the information in the File System Document Export action section to configure the DOCUMENT_ID, ExportDirectory, Separator, and extra file name segment parameters, as needed.

   Note: Keep your file system file name character limits in mind as you configure the file name parameters. You can click **Save Inputs** to save the input mapping. You can also click **Validate inputs** to validate the input mapping.

8. Click **Enable channel**. When you are prompted to enable the channel, click **OK**.

   Note: After you create a channel, you cannot modify the configuration.

Create PIC Status Update channels

PIC Status Update channels route documents in Perceptive Content workflow as they pass through different stages of Intelligent Capture processing. The channels from version 1.2.5 are not compatible with version 2.0.2, however a part of the XML is identical..

These channels use the PIC project name, client ID, and the status code to route documents to specified queues. For example, you can create a channel that routes documents that failed the extraction process, such as status code 250, to a queue called IC Processing Error. Your users can then review documents in the queue and resubmit them for processing.

You can create one channel per combination of project name, client ID, and status code.

You cannot replace or modify an enabled channel. For assistance with modifying an Intelligent Capture Status Update channel, contact your Perceptive Software representative.

Note: Before you begin configuring a channel, we recommend that you open the *Perceptive Connect Runtime Installation Online Help* to use as reference material. For information on the client ID, refer to the "BRWClient" section in the *Perceptive Intelligent Capture for Invoices Solution Guide 2.3.x* or contact your administrator for other Intelligent Capture solutions.

Channel requirements

A PIC Status Update channel requires the following components.

- A PIC project name, client ID, and status code
- Two Perceptive Content queue names
- The Status Update trigger, provided by this connector
- The RouteImageNowWorkflowItem action, provided by the Perceptive Content Connector

Create and configure a PIC Status Update channel

To create and configure a Status Update channel, complete the following steps.

1. In a browser, navigate to the channel wizard URL in the format `http://<Perceptive Connect Runtime machine>:<port>`.
2. In the left pane, under **Manage**, click **Create a Channel**.
3. Optional. In the right pane, in the **Channel** section, enter a name and description for the channel.

   Note: Entering a name and description for the channel may help you find the channel on the **List Channels** page.

4. In the **Trigger Information** section, complete the following substeps.
   1. In the **Trigger** list, select **Status Update Trigger**.
   2. In the **Project Name** box, type the name of the PIC project that processes the documents you want to route, such as `DFI-BPO`. Do not include the file extension.

      Note: The PIC project name is set in the `GRL_VL_ProjectName` setting in the `PIC DFI.ini`

file.

3. In the **Client ID** box, type the PIC client ID. Client IDs are non-negative integers.

4. In the **Status Code** box, type the PIC status code that you want this channel to handle, such as 550. Status codes are numbers between 0-999.

5. Click **Continue**.

5. On the **Modify Channel Mapping** page, under **Actions** , select **RouteImageNowWorkflowItem**.

6. From the temporary directory where you downloaded the connector files, open **StatusUpdateChannel_ InputMapping.txt** file in the **RouteImageNowWorkflowItem** folder to use as configuration examples.

7. To configure the inputs, in the **Inputs** XML text editor, use the sample XML files and the information in the Route ImageNow Workflow Item action section to configure the `WorkflowItemID`, `SuccessQueueName` and `FailureQueueName`.

> **Note:** You can click **Save Inputs** to save the input mapping. You can also click **Validate inputs** to validate the input mapping. You cannot configure any writer outputs for a PIC Status Update channel.

8. When prompted to enable the channel, click **OK**.

Create PIC Export channels

PIC Export channels pass XML data, extracted by Intelligent Capture, back to your original documents in Perceptive Content. With a PIC Export channel, you can write this XML data to document keys, custom properties, eForm fields, or the DCExport subobject. The channel uses the XML transformer, a specialized reader, to migrate the extracted data to an XML template or schema. The channels from version 1.2.5 are not compatible with version 2.0.2, however a part of the XML is identical.

These channels use the PIC project name, client ID, and document class to identify which data to export from PIC to Perceptive Content. For example, you can create a channel that uses extracted information from the image to populate an eForm or update indexes in Perceptive Content.

> **Note:** You cannot replace an enabled channel. For assistance with modifying an Intelligent Capture Export channel, contact your Perceptive Software representative

For information on the PIC client ID, refer to the "BRWClient" section in the *Perceptive Intelligent Capture for Invoices Solution Guide 2.7.x* or contact your administrator for other Intelligent Capture solutions.

Channel requirements

A PIC Export channel requires the following components.

- A PIC project name, client ID, and document class
- The Document trigger, DCExport Subobject writer, and Document Property writer provided by this connector
- The XML reader and XML transformer, provided by the Perceptive Connect Runtime
- The RouteImageNowWorkflowItem action, eForm Data Definition reader, and eForm writer provided by the Perceptive Content Connector

Create and configure a PIC Export channel

This section provides a high-level overview of how to create PIC Export channels. When you create a PIC Export channel, you must configure the input and output XML data to map the extracted Intelligent Capture data to your specific data definition or schema. This process is highly variable depending on your target. As you read this overview, we recommend that you refer to the detailed appendices for the required components.

To create and configure a PIC Export channel, complete the following steps.

1. In a browser, navigate to the channel wizard URL in the format `http://<Perceptive Connect Runtime machine>:<port>`.
2. In the left pane, under **Manage**, click **Create a Channel**.
3. Optional. In the right pane, in the **Channel** section, enter a name and description for the channel.

   > **Note:** Entering a name and description for the channel may help you find the channel on the **List Channels** page.

4. In the **Trigger Information** section, complete the following substeps.
   1. In the **Trigger** list, select **Document Trigger**.
   2. In the **Project Name** box, type the name of the Intelligent Capture project that processes the documents you want to route, such as `DFI-BPO`. Do not include the file extension.

      > **Note:** The PIC project name is set in the `GRL_VL_ProjectName` setting in the `PIC DFI.ini` file.

   3. In the **Client ID** box, type the Intelligent Capture client ID. The client ID consists of a non-negative integer.
   4. In the **Document Class** box, type the Intelligent Capture parent document class, such as `Invoices` or `Void`, and click **Continue**.
   5. In the **Mode** box, type `EXPORT`.
5. On the **Modify Channel Mapping** page, under **Actions**, select **RouteImageNowWorkflowItem**.
6. From the temporary directory where you downloaded the connector files, open the **Export_Invoice_ Inputs.xml, Export_Void_Inputs.xml**, and **Export_Invoice_Outputs.xml** files in the **RouteImageNowWorkflowItem** folder.

   > **Note:** These files demonstrate how to map extracted data to the Perceptive AP Invoice eForm data definition. If you are mapping values to a different target, use these files as a general example of the overall process.

7. In the **Inputs** XML text editor, configure the data sources with the following substeps. These steps assume you will map values to the AP Invoice eForm, and will vary for other targets.
   1. Configure the `WorkflowItemID`, `SuccessQueueName`, and `FailureQueueName`.
   2. Configure the XML reader to extract the invalid reason code, **IC_InvalidReason**, from the PIC XML. You need the Xpath to the invalid reason value in the PIC XML.
   3. Configure the XML reader to extract additional values from the PIC XML, such as `InvoiceType` and `InvoiceNumber`, which you will use when you map the outputs.

4. Configure the eForm Data Definition reader, `eFormDataDefinitionSource`, to extract the data definition from your form, such as AP Invoice eForm.

5. Configure the XML transformer to map the extracted PIC values to your form's data definition. This maps the additional values extracted from the PIC XML to the data definition extracted from your form.

> **Note:** You can click **Save Inputs** to save the input mapping. You can also click **Validate inputs** to validate the input mapping.

8. Click the **Outputs** tab.

9. In the **Outputs** XML text editor, configure the outputs with the following substeps.

   1. Configure the eForm writer to write the transformed XML values into your eForm fields. The writer references the name of the XML transformer. In the **Export_Invoice_Inputs.xml** file, this is `TransformedXml`.

   2. Configure the Document Property writer to write the `IC_InvalidReason` code to the `IC Invalid Reason` custom property.

   3. Optional. Configure the DCExport Subobject writer to write the PIC XML, `ic_xml`, to Perceptive Content's DCExport Subobject.

> **Note:** You can click **Save Outputs** to save the output mapping. You can also click **Validate outputs** to validate output mapping.

10. In the `<c:outputs ... />` element, remove the trailing backslash (/). Then, after the final `</c:parameter>` value, add a closing `</c:outputs>` element.

11. Click **Enable channel**. When the UI prompts you to enable the channel, click **OK**.

Triggers

A trigger is an event that causes a channel to execute. Each channel has a single trigger, so the event defined by the trigger and its inputs is the only entry point into a given channel. Triggers can also bring data into the channel. This data is available for input mapping for the channel action or for the results output. For additional information about triggers, refer to the "Appendix A: Triggers" section of the *Perceptive Content Connector Installation Guide*.

All of the triggers for the Perceptive Capture Connector use REST calls. Each trigger includes output values that you can use in the channel data context for either input or output mappings.

The Perceptive Capture Connector provides the following triggers and associated data.

- Document trigger
- Status Update trigger

Document trigger

Document Trigger allows you to create channels for creating and soft deleting documents in Perceptive Content. You can also use this trigger to create channels to pass XML content back to documents in Perceptive Content.

This trigger includes the following input identifiers.

- Project Name. The PIC Project Name.
- Client Id. The client ID configured in PIC.
- Document Class. The document class configured in PIC. If no document class is configured in PIC, configure the document class with "GLOBAL".
- Mode. The three possible modes are CREATE, EXPORT and DELETE

In CREATE mode the following data is made available to the channel by the trigger.

- A standard document image with the key STANDARD_IMAGE
- An optional original image with the key ORIGINAL_IMAGE
- A metadata XML with the key XML_DOC

In EXPORT mode only the Perceptive Content document ID with the key DocumentID is made available to the channel by the trigger, in addition to the data provided by the CREATE mode. This data can be read using the Trigger Reader provided by Perceptive Connect Runtime.

In DELETE mode only the Perceptive Content document ID with the key DocumentID is made available to the channel by the trigger.

Status Update trigger

The Status Update trigger allows you to create Status Update channels for PIC.

This trigger includes the following input identifiers.

- ProjectName. The PIC Project Name.
- ClientId. The client ID configured in PIC.
- StatusCode. The PIC status code which is a non-negative integer between 0-999.

Use trigger values in a channel input or output mapping

To use a Status Update trigger value in the input mapping, use the Perceptive Content Connector Trigger reader. In this instance, the trigger node contains the output value from the selected trigger and the trigger name is any string that you use as a reference in another node.

Example

```
<c:parameter>
      <c:name>DocumentId</c:name>
      <c:trigger>DOCUMNET_ID</c:trigger>
</c:parameter>
```

To use a trigger value in the output mapping, such as for the DC Export Sub-object, reference the value directly.

Example

```
<c:parameter>
        <c:name>ICXmlDoc</c:name>
        <in:DCExportTarget>DocumentId</in:DCExportTarget>
</c:parameter>
```

Actions

An action is a connector-defined task configured in the channel that executes when the channel triggers. You can configure the various inputs that the action requires when you create the channel. The action uses these inputs to complete its task in an application outside of Perceptive Connect Runtime, as defined in the connector. For additional information about actions, refer to the "Appendix B: Actions" section of the *Perceptive Content Connector Installation Guide*.

The Perceptive Capture Connector provides the following actions.

- Capture Document action
- File System Document Export action

Capture Document action

The Capture Document Action allows you to create a document in Perceptive Content using the configured Capture Profiles and Application Plans. After receiving an image and metadata for document keys and custom properties, the Capture Profile name, and other configuration parameters, this action creates a document with all the information in Perceptive Content and routes it to the workflow queue configured in the Capture Profile.

When you select the Capture Document action, Perceptive Connect Runtime automatically populates the following XML configuration template in the Inputs box.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:inputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>StandardImage</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>OriginalImage</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>CaptureProfileData</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
```

```
        <c:name>CaptureProfile</c:name>
        <c:none/>
    </c:parameter>
</c:inputs>
```

File System Document Export action

This action exports any Perceptive Content documents from an Integration ASQ to a specific network location, such as the Intelligent Capture import directory.

With the File System Document Export action, you can perform the following actions.

- File name format configuration. You can include Perceptive Content values or literal values in the file naming convention and pass it to Perceptive Intelligent Capture.
- File splitting. You can split a multi-page document into smaller documents and export these as separate files to the file system.
- File merging. You can merge multiple documents into a single document.
- TIFF conversion. If you have multiple documents with variable format, you can convert all the documents into TIFF format before exporting the documents.

When you select the File System Document Export action, Perceptive Connect Runtime automatically populates the following XML configuration template in the Inputs box.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:inputs xmlns:ns2="http://www.perceptivesoftware.com/pie/mapping/sample"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>WfId</c:name>
        <c:trigger>WorkflowItemId</c:trigger>
    </c:parameter>
    <c:parameter>
        <c:name>DOCUMENT_ID</c:name>
        <in:workflowItem
xmlns:in="http://www.perceptivesoftware.com/pif/imagenow">
            <in:reference>WfId</in:reference>
            <in:objectType>WORKFLOW</in:objectType>
            <in:objectField>OBJECT_ID</in:objectField>
            </in:workflowItem>
    </c:parameter>
    <c:parameter>
        <c:name>ExportDirectory</c:name>
        <c:literal>E:\Temp</c:literal>
    </c:parameter>
    <c:parameter>
        <c:name>Separator</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>EnableSubDirectory</c:name>
```

```
            <c:literal>true</c:literal>
        </c:parameter>
        <c:parameter>
            <c:name>SinglePageDocFlag</c:name>
            <c:literal>true</c:literal>
        </c:parameter>
        <c:parameter>
            <c:name>Convert</c:name>
            <c:none/>
        </c:parameter>
        <c:parameter>
            <c:name>Segment1</c:name>
            <c:literal>S1</c:literal>
        </c:parameter>
```

The action requires that you specify the `DOCUMENT_ID` and `ExportDirectory` parameters. `DOCUMENT_ID` retrieves the ID of the Perceptive Content document that the action is exporting. The ID is always the first element of the file name.

The Integration ASQ Trigger provides the Doc ID. However, if you use this action with another trigger that does not provide the Doc ID, the Workflow Item reader can use the workflow ID to retrieve the Doc ID.

Parameter Descriptions

`Convert` allows you to specify whether to process a document in the original format or TIFF format. To process a document as TIFF, add TRUE to the literal reader or add FALSE to keep the existing format. However, if a Perceptive Content document has both, PDF and TIFF files, and the value of the SinglePageDocFlag parameter is set to FALSE, PCC Connector converts the PDF files to TIFF irrespective of whether you specify TRUE or FALSE value to this parameter.

`EnableSubDirectory` enables you to configure the connector to create a subdirectory in the output directory for the processed Perceptive Content document. To create a subdirectory in the export directory add TRUE to the literal reader.

`SinglePageDocFlag` enables you to configure the connector to create individual files for each page that makes up a Perceptive Content document. To create individual files for each page of a file add TRUE to the literal reader. For example, if one TIFF file and one PDF file with ten pages each are associated with a Perceptive Content document, ten PDF files and ten TIFF files are created separately in the export directory if you set the literal reader for `SinglePageDocFlag` as TRUE. The files created inherit their native file type.

`ExportDirectory` sets the output location on the file system. If the export directory is on a network location, the Perceptive Connect Runtime user must specify the network path of the export directory under the `ExportDirectory` parameter. To specify the export directory, replace `<c:none/>` with an empty set of literal reader from Perceptive Connect Runtime, and provide the network path value. The following example describes the `ExportDirectory` parameter.

```
<c:parameter>
  <c:name>ExportDirectory</c:name>
  <c:literal>//192.168.45.101/Shared/Export</c:literal>
</c:parameter>
```

> **Note:** The Perceptive Connect Runtime user must have write access to the export directory.

`Separator` sets the separator character for the file name. If you do not specify a character, the default is `^`. Refer to the following list for acceptable separator characters.

```
~ ! @ # $ % ^ & ( ) - _ + [ ] { } ; ' , .
```

With the `Segment` parameters, you can configure up to 99 additional segments in the file name. Keep in mind your system file name character limits.

To insert a Perceptive Content document key value, custom property value, or Perceptive Content document name, in the `Segment` parameters, replace `<none\>` with the element `<objectPropertyReader>`. The element `<objectPropertyReader>` is described below.

```
<in:objectPropertyReader>
  <in:name>Field4</in:name>
  <in:objectIdRef>DOCUMENT_ID</in:objectIdRef>
  <in:objectType>document</in:objectType>
  <in:propertyType>KEY</in:propertyType>
</in:objectPropertyReader>
```

Modify `<objectPropertyReader>` using any of the following options.

- To insert a literal string, use the literal reader from Perceptive Connect Runtime. For example, insert a company name or business unit. To insert an empty segment, use the literal reader from Perceptive Connect Runtime with an empty set.

```
<c:literal></c:literal>
```

> **Note:** You cannot use the characters `< > : " / \ | ? *` in the Segment values or as the separator.

- To insert a Perceptive Content document key value, in the element `<in:propertyType>`, type `KEY`, and in the element `<in:name>`, specify a key value. Allowed values are `field1`, `field2`, `field3`, `field4`, `field5`, `folder`, `tab`, `drawer`, and `type`.
- To insert a Perceptive Content custom property, in the element `<in:propertyType>`, type `CUSTOMPROPERTY`, and in the element `<in:name>`, specify the name of the custom property.
- To insert the Perceptive Content document name, in the element `<in:propertyType>`, type `DOCPROPERTY`, and in the element `<in:name>`, specify the document name.

```
<c:parameter>
    <c:name>DOCUMENT_ID</c:name>
    <in:workflowItem>
        <in:reference>WfId</in:reference>
        <in:objectType>WORKFLOW</in:objectType>
        <in:objectField>OBJECT_ID</in:objectField>
    </in:workflowItem>
</c:parameter>
<c:parameter>
    <c:name>WfId</c:name>
```

```
        <c:trigger>WorkflowItemId</c:trigger>
    </c:parameter>
    <c:parameter>
        <c:name>ExportDirectory</c:name>
        <c:literal>C:\Program Files\Perceptive\Projects\AP\Export</c:literal>
    </c:parameter>
    <c:parameter>
        <c:name>Separator</c:name>
        <c:literal>#</c:literal>
    </c:parameter>
    <c:parameter>
        <c:name>Segment1</c:name>
        <c:literal>ID</c:literal>
    </c:parameter>
    <c:parameter>
        <c:name>Segment2</c:name>
        <in:objectPropertyReader>
            <in:name>Field4</in:name>
            <in:objectIdRef>DOCUMENT_ID</in:objectIdRef>
            <in:objectType>document</in:objectType>
            <in:propertyType>KEY</in:propertyType>
        </in:objectPropertyReader>
    </c:parameter>
```

Note: The system user must have write access permission to the ExportDirectory location you mention.

Writers

Writers are interfaces, provided by connectors, which let you configure channels to output data to applications outside of Perceptive Connect Runtime. You use writers to configure the results output mapping, which lets you specify what to do with any data resulting from the action execution. You can invoke writers using specific XML tags, defined per writer. For additional information about writers, refer to the "Appendix D: Writers" section of the *Perceptive Content Connector Installation Guide*.

The Perceptive Capture Connector provides the following writers.

- DCExport Subobject writer
- Document Property writer
- Image writer

DCExport Subobject writer

This writer lets you save an XML document (w3c.Document Java type) to the DCExport subobject of a document in Perceptive Content. Perceptive Content uses the workingName parameter to identify objects attached to the DCExport subobject. The workingName value is the root node from the XML document to be stored, such as BrainwareDocument. If an object with the same workingName already exists, the writer replaces the data with the latest XML document.

To output an XML document from a channel to a Perceptive Content document, include the following XML

template in the channel output mapping, and enter the appropriate values.

```
<c:name></name>
```

```
<in:DCExportTarget></in:DCExportTarget>
```

The `name` field references the name node of the Perceptive Content Connector Trigger reader. Refer to the "Trigger" section in the *Perceptive Connect Runtime Installation and Setup Guide*.

In the `DCExportTarget` field, reference the Perceptive Content document ID to which you want to attach the XML. For example, if you used the Status Update trigger from this connector, you can use the trigger's output value `DocumentId`. If you used the Integration ASQ Trigger, from the Perceptive Content Connector, you would reference the Workflow Item reader name, such as `DOCUMENT_ID`.

DCExport subobjects only exist at the document level. You cannot write this subobject to a Perceptive Content folder, or to an individual page within a Perceptive Content document.

Example

```
<c:parameter>
      <c:name>ICXmlDoc</c:name>
      <in:DCExportTarget>DocumentId</in:DCExportTarget>
</c:parameter>
```

Document Property writer

The Document Property writer enables you to output a value to a document property in Perceptive Content.

To write a value from the channel to a document property, include the following XML template in the channel output mapping, with the appropriate values.

```
<in:documentPropertyWriter>
      <in:keyMappings>
            <in:keyMapping>
                  <in:source></in:source>
                  <in:target></in:target>
            </in:keyMapping>
      </in:keyMappings>
      <in:customPropertyMappings>
            <in:customPropertyMapping>
                  <in:source></in:source>
                  <in:target></in:target>
            </in:customPropertyMapping>
      </in:customPropertyMappings>
      <in:errorCustomProperty></in:errorCustomProperty>
      <in:ignoreMissingProperty></in:ignoreMissingProperty>
</in:documentPropertyWriter>
```

When configured, the `keyMappings` section must have minimum one and maximum seven unique `keyMapping` values. The `source` field contains a reference to the value to be stored against the property. The `target` field contains the key name and is restricted to `FIELD1`, `FIELD2`, `FIELD3`, `FIELD4`, `FIELD5`, `TYPE` and `DRAWER`.

When configured, the `customPropertyMappings` section can have one or more `customPropertyMapping` values. The `source` field contains a reference to the value to be stored against the property. The `target` field contains the name of the custom property to be populated.

The `errorCustomProperty` section contains the name of a custom property to which you can write error messages when an error occurs while updating document keys and custom properties. Alternatively you can leave the field blank if you don't want to write error messages.

The `ignoreMissingProperty` section contains a TRUE or FALSE value. When this value is set to TRUE, the writer logs a warning message that the custom property could not be found and continues processing any unprocessed custom properties. When this value is set to FALSE, the writer shows an error that the property could not be found and any unprocessed custom properties are not processed. When this property is not present in the configuration, it is set to FALSE by default.

Custom Properties

Perceptive Content identifies the custom property type. The type need not be passed to the writer, but the writer internally translates values in the context to String for the STRING, NUMBER, DATE and FLAG types and the String should be in a specific format. The following custom property types are acceptable.

STRING This is stored in the custom property just as it exists in the context.

NUMBER The number is parsed. If the string is not a number, the writer will fail.

DATE The date must be in the format MM/DD/YYYY.

FLAG The following values will be parsed into the corresponding flag value. Any other value will cause the writer to fail.

- 0 False
- 1 True
- false False
- true True

The writer does not support writing to custom properties of types "USER_GROUP", "USER_LIST", "COMPOSITE" and "ARRAY".

Examples:

In the below sample mapping, the writer updates both document keys and custom properties.

```
<c:parameter>
        <c:name>DOCUMENT_ID</c:name>
        <in:documentPropertyWriter>
                <in:keyMappings>
                <in:keyMapping>
                        <in:source>InvoiceNumber</in:source>
```

```
                <in:target>FIELD1</in:target>
        </in:keyMapping>
        <in:keyMapping>
                <in:source>VendorName</in:source>
                <in:target>FIELD2</in:target>
        </in:keyMapping>
        <in:keyMapping>
                <in:source>PONumber</in:source>
                <in:target>FIELD3</in:target>
        </in:keyMapping>
        </in:keyMappings>
        <in:customPropertyMappings>
        <in:customPropertyMapping>
                <in:source>VendorNumber</in:source>
                <in:target>Vendor ID</in:target>
        </in:customPropertyMapping>
        <in:customPropertyMapping>
                <in:source>VendorName</in:source>
                <in:target>Vendor Name</in:target>
        </in:customPropertyMapping>
        <in:customPropertyMapping>
                <in:source>InvoiceDate</in:source>
                <in:target>Invoice Date</in:target>
        </in:customPropertyMapping>
        </in:customPropertyMappings>
        <in:errorCustomProperty>Error</in:errorCustomProperty>
        <in:ignoreMissingProperty>true</in:ignoreMissingProperty>
    </in:documentPropertyWriter>
    </c:parameter>
```

In the below sample mapping, the writer is configured to update only document custom properties.

```
<c:parameter>
    <c:name>DOCUMENT_ID</c:name>
    <in:documentPropertyWriter>
                        <in:customPropertyMappings>
        <in:customPropertyMapping>
                <in:source>VendorNumber</in:source>
                <in:target>Vendor ID</in:target>
        </in:customPropertyMapping>
        <in:customPropertyMapping>
                <in:source>VendorName</in:source>
                <in:target>Vendor Name</in:target>
        </in:customPropertyMapping>
        <in:customPropertyMapping>
                <in:source>InvoiceDate</in:source>
                <in:target>Invoice Date</in:target>
```

```
            </in:customPropertyMapping>
            </in:customPropertyMappings>
            <in:errorCustomProperty>Error</in:errorCustomProperty>
            <in:ignoreMissingProperty>true</in:ignoreMissingProperty>
        </in:documentPropertyWriter>
    </c:parameter>
```

In the below sample mapping, the writer is configured to update only document keys.

```
<c:parameter>
        <c:name>DOCUMENT_ID</c:name>
        <in:documentPropertyWriter>
                <in:keyMappings>
                <in:keyMapping>
                        <in:source>InvoiceNumber</in:source>
                        <in:target>FIELD1</in:target>
                </in:keyMapping>
                <in:keyMapping>
                        <in:source>VendorName</in:source>
                        <in:target>FIELD2</in:target>
                </in:keyMapping>
                <in:keyMapping>
                        <in:source>PONumber</in:source>
                        <in:target>FIELD3</in:target>
                </in:keyMapping>
                </in:keyMappings>
                <in:errorCustomProperty>Error</in:errorCustomProperty>
                <in:ignoreMissingProperty>true</in:ignoreMissingProperty>
        </in:documentPropertyWriter>
    </c:parameter>
```

Image writer

The Image writer enables you to split a multi-page document, convert it to TIFF, and replace it with the existing pages in a Perceptive Content document.

You can either provide a new image as input on which the operations are performed or the image can be fetched from Perceptive Content based on an input Document ID. You can also add an 'original' file or image to the document as its last page. You cannot perform any split or conversion operations on this 'original' image or file.

To perform the above operations add the following XML template to the output mapping of a channel.

```
<in:imageWriter>
    <in:replaceWith></in:replaceWith>
    <in:convertToTiff></in:convertToTiff>
    <in:splitMultiPageTiff></in:splitMultiPageTiff>
    <in:original></in:original>
</in:imageWriter>
```

The `replaceWith` field contains a reference to the image on which you perform the split and conversion operations. If you leave this field blank, the image is fetched from Perceptive Content and all operations are performed on this fetched image.

The `convertToTiff` field contains a TRUE or FALSE value. Setting this value to TRUE converts any image file format to TIFF.

The `splitMultiPageTiff` field contains a TRUE or FALSE value. Setting this value to true splits a multi-page file into individual files, each consisting of a single page. The resulting files are TIFF files, irrespective of whether `convertToTiff` is set to TRUE or FALSE.

The `original` field contains a reference to an image or a file that you would like to add as the last page of a document.

Example

```
<c:parameter>
    <c:name>DOCUMENT_ID</c:name>
    <in:imageWriter>
        <in:replaceWith>StandardImage</in:replaceWith>
        <in:convertToTiff>true</in:convertToTiff>
        <in:splitMultiPageTiff>true</in:splitMultiPageTiff>
        <in:original>OriginalImage</in:original>
    </in:imageWriter>
</c:parameter>
```

Here, the writer splits `StandardImage` (a multi-page image file in any format) into multiple TIFF files each consisting of a single page. The existing pages in the Perceptive Content document get replaced with these new files. The `OriginalImage` file is added as the last page of the document.

The `StandardImage` and `OriginalImage` files are usually made available in the Perceptive Capture Connector channel by Document Trigger and can be read using Perceptive Capture Connector Trigger Reader.

```
<c:parameter>
    <c:name>DOCUMENT_ID</c:name>
    <in:imageWriter>
        <in:replaceWith></in:replaceWith>
        <in:convertToTiff>true</in:convertToTiff>
        <in:splitMultiPageTiff>true</in:splitMultiPageTiff>
        <in:original></in:original>
    </in:imageWriter>
</c:parameter>
```

In this example, the writer fetches the images from the Content document, performs the operations on these images, and replaces the old pages in Perceptive Content with the new pages. This writer does not add an original image as the last page as the configuration is not set.

When used in a late archiving scenario, in the event of duplicate indexes, this configuration retains all existing pages in Perceptive Content. However, the original image from previous conflicting create calls may be split or converted to TIFF based on the configuration. The original image from the last create call remains unaffected.

Example of Perceptive Content workflow for Perceptive Intelligent Capture

The following image provides an example layout of the Perceptive Content queues provided for the Perceptive Intelligent Capture workflow. If you create a File Output channel, for example, you would use the IC Submit Integration ASQ to trigger the channel if you have Perceptive Content 7.2 or earlier installed on your machine.
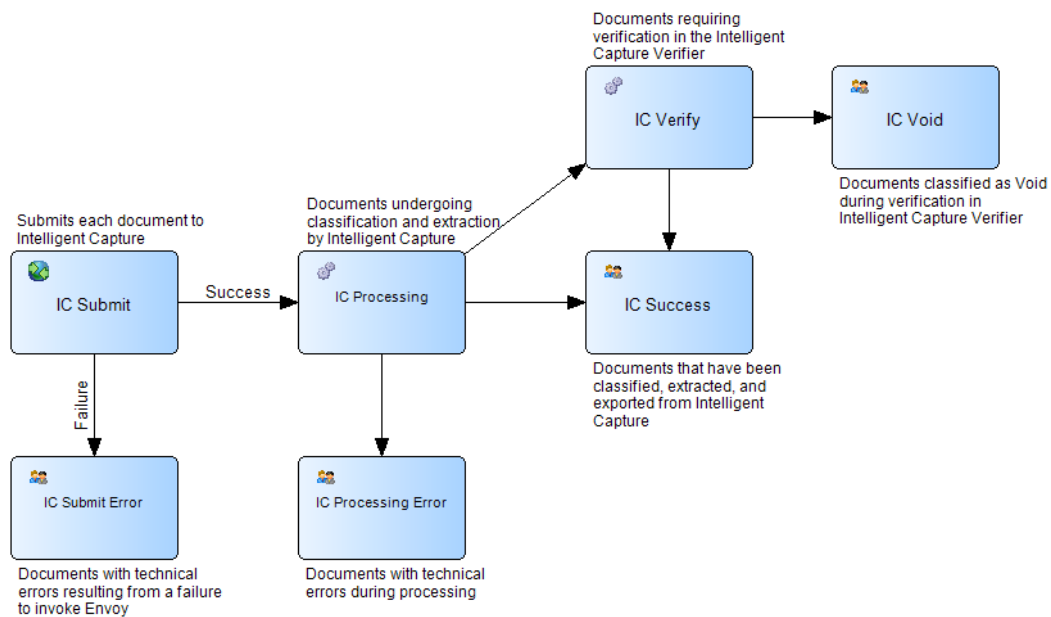
If you have Perceptive Content 7.2 installed on your machine, you would use the IC Submit Connect ASQ to trigger the channel.

Documents requiring
verification in the Intelligent
Capture Verifier

IC Verify → IC Void

Documents classified as
void during verification in
Intelligent Capture Verifier

Submits each document to
Intelligent Capture

Documents undergoing
classification and extraction
by Intelligent Capture

IC Submit — Success → IC Processing → IC Success

Documents that have been
classified, extracted, and
exported from Intelligent
Capture

Failure

IC Submit Error   IC Processing Error

Documents with technical
errors resulting from a
submit failure

Documents with technical
errors during processing