Application Plan Designer

User Guide

Version: 7.4

Written by: Documentation Team, R&D
Date: Friday, February 13, 2026

HYLAND™

Documentation Notice

Table of Contents

What is Application Plan Designer?

In Application Plan Designer, you create application plans mapped to your business application or for collecting data.

The following bullets highlight the components you define in the Application Plan Designer.

- Screens. Allows you to define screens captured from your business application for a LearnMode type application plan.
- Dictionary. Allows you to define elements captured for an application plan. You can capture an unlimited amount of metadata, as well as associate dictionary data elements to document properties, custom properties, and folder properties.
- Elements. Allows you to locate and identify the data elements you want to add to the dictionary. Depending on the method, the Elements tab displays the bitmap (HyperLearn or Viewpoint), text (Terminal Capture or DOS), table (Internet Explorer), or tree (Window Walker) related to the business application screen you are currently trying to learn.
- Map. Allows you to define document, folder, and shortcut properties when using an application plan.
- View Action. Allows you to define the documents or folders that the system displays in ImageNowExplorer. You can configure View Action to return folders or documents.

What is LearnMode?

LearnMode is a client-side, non-programmatic method of integrating your business application with Perceptive Content and learning screens or windows within that application.

The LearnMode type application plan offers six different methods to learn your business application, and the best method to use depends on the type of application you are using. Knowing whether your application is GUI-based, web-based, DOS-based, a terminal emulator, or other type of application will assist in determining the LearnMode method to use. Sometimes, it takes trying more than one LearnMode method to determine which one works best for your particular application.

Each application plan you create can use a different LearnMode method. A LearnMode application plan contains screens. Creating a screen involves capturing the data from your business application screen so that you can create properties to link the image to a business object in your application. There should be one screen in the application plan for each screen or window in your business application. For example, a General Banking System business application may have a Main screen, a Summary screen, and a Loan screen.

Get started with Application Plan Designer

Customize Application Plan Designer

In Application Plan Designer, you create application plans mapped to items. To customize the look of Application Plan Designer, complete the following steps.

1. In **Application Plan Designer**, click **File** > **Options**.
2. In **Application Plan Designer Options** dialog box, do any of the following to customize your **Application Plan Designer**.
   - From the **Bitmap selection color** box, modify the RGB values if you want to customize the color Perceptive Content uses when selecting the business application field from the bitmap.
   - In the **Table defined color** box, modify the RGB values if you want to customize the color Perceptive Content uses.
   - In the **Text defined color** box, modify the RGB values if you want to customize the color Perceptive Content uses.
   - In the **Minimize on dictionary test** list, select **True** if you want Perceptive Content to minimize **Application Plan Designer** window when testing the data dictionary for the application plan.
3. Click **Apply** and then **OK** to close the dialog box.

Shortcut keys in Application Plan Designer

Many Perceptive Content functions can be performed using shortcut keys. Shortcut key functions vary depending on which grid or component of Perceptive Content you are using.

Shortcut	Description
+	Zoom In
Ctrl + S	Save
Ctrl + X	Cut
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + D	Capture Data Field
Ctrl + T	Test
DEL	Delete

Shortcut	Description
F1	Help
F5	Refresh
NUM -	Zoom Out

Application Plan Designer toolbars

The following toolbars are available in Application Plan Designer.

Application Plan toolbar

Button	Name	Description
	Plan Settings	Displays the Plan Settings dialog box, which lets you modify application plan and security settings.
	Save	Saves the application plan.
	Options	Displays the Application Plan designer Options dialog box, which lets you set the properties for the application plan.

Dictionary toolbar

Button	Name	Description
	New Field	Displays the Dictionary Field dialog box, which lets you add a data element to the dictionary.

Button	Name	Description
	Delete Field	Deletes the selected data element from the dictionary.
	Remove from Current Screen	Removes the selected data element from the current screen.
	Modify Field	Modifies the selected data element.
	Move Up	Moves the selected data element up.
	Move Down	Moves the selected data element down.
	Test Dictionary	Displays the Test Results dialog box, which lets you test the data dictionary for the application plan.

Map toolbar

Button	Name	Description
	Expand Map	Expands all map levels.
	Collapse Map	Collapses all map levels.
	Test Map	Displays the Test Results dialog box, which lets you test the application plan map.

Map Levels toolbar

Button	Name	Description
	Add folder level	Creates a folder level in the map

Button	Name	Description
	Add document level	Adds a document level to a folder map. If you add a document level, the map type changes to a document map.
	Add shortcut map	Creates a shortcut for the document or folder in the map.
	Modify Row	Modifies the selected property.
	Move Down	Moves the selected level down.
	Move Up	Moves the selected level up.
	Removes Level	Removes the selected map level.
	Reset Level	Resets the values in the selected level to default settings.
	Reset Row	Resets the value in the selected row to a default setting.
	Linked Content	Appears where linked content is in use.
	Alert	Appears where you need to define or modify values.
	Required Custom Property	Appears when a custom property defined as required for the specified folder type or document type is in use.

Screen Elements toolbar

Button	Name	Description
	Capture Screen Identifier	Updates the selected screen s identifier using the data element that is currently selected in the capture window. If no data element is selected, ImageNow uses the business application s window title as the identifier.
	Capture Data Field	Displays the Dictionary Field dialog box, which lets you add a data element to the dictionary.
	Expand All (Window Walker)	Expands the tree structure in the capture window.
	Zoom In (HyperLearn)	Magnifies the content in the capture window by a factor of 2.
	Increase Font (Terminal Capture and DOS command prompt)	Increases the font size in the capture window.
	Collapse All (Window Walker)	Collapses the tree structure in the capture window.
	Zoom Out (Window Text and HyperLearn)	De-magnifies the content in the capture window by a factor of 2.
	Decrease Font (Terminal Capture and DOS command prompt)	Decreases the font size in the capture window.
	Find	Finds text within the contents of the capture window.
	Enable HyperLearn Debug	Enables HyperLearn debug, which displays the HyperLearn dialog box when a data element is created.
	Enable Point Dialog	Enables Point dialog, which lets you update the screen's identifier. Point dialog is used when the location of data on the business application screen changes slightly upon each use.

Screens toolbar

Button	Name	Description
	New Screen	Displays the Screen Properties dialog box, which lets you add a screen from your business application to your application plan.
	Duplicate Screen	Duplicates the selected screen.
	Delete Screen	Deletes the selected screen.
	Move Up	Moves the selected screen up.
	Move Down	Moves the selected screen down.

Troubleshoot application plan testing

If you encounter error messages while testing your application plan or application plan map, try any of the following possible resolutions.

I receive an error when I test my application plan map

Error	Resolution
Required value	Verify whether required values have been left blank. To enable the Map test, enter a value where required, or modify your application plan as needed.
The folder type does not match the type set for the folder structure. Select a different folder type.	Verify whether the folder type is valid by accessing the folder type hierarchy in Management Console.
The document type does not match the type set for the folder structure. Select a different document type.	Verify whether the document type is valid in the hierarchy specified for the drawer in Management Console.

ImageNow cannot find my application plan screen

Cause	Resolution
The appropriate screen is not being displayed within the business application.	Switch to the appropriate screen within the business application and test again.
The screen identifier used to create the application plan screen is not present.	Check the screen identifier within LearnMode. You might need to rebuild the application plan screen using a different screen identifier.
The screen identifier being used is not static and/or not unique to the current business application screen.	Check the screen identifier within LearnMode. You might need to rebuild the application plan screen using a different screen identifier.

ImageNow cannot locate my business application

Cause	Resolution
The business application is not running.	Start the business application and go to the appropriate screen.
The name of the application as it appears in the title bar of the business application window has changed since creating the application plan.	Open the LearnModeapplication plan and verify that the window name matches the name of the business application window. You might need to modify this value.
The name of the business application window may change for each user or for each login session.	Some applications allow you to modify the title of the window. Check to see if such settings exist within the business application or if you can use a wildcard.

ImageNow cannot retrieve data from my clipboard

Cause	Resolution
The appropriate connect strings are not being used.	Open the LearnModeapplication plan and verify that the correct connect strings are being used. Be sure that each connect string segment ends with a semicolon (;). Verify that the connect strings you are using are valid.
The operation that attempts to copy the data from	Make changes to the Retry/Delay settings to change

Cause	Resolution
the business application is happening too fast.	the timing of the data capture.
The terminal emulation software being used to connect to the business application is not the same as the emulator used when creating the application plan.	Make sure that the same terminal emulation software is being used on each computer running ImageNow Client. Be sure that all the versions are the same; different versions will sometimes behave differently.

The capture profile and application settings conflict

Cause	Resolution
The settings in the capture profile and the selected application plan specify to store the captured files in different locations.	Select an application plan that does not conflict with the capture profile.
The capture profile specifies to store the captured files in a folder, but the selected application plan specifies to store the files in a different folder or in a drawer, or the application plan stores the document in a folder, but the capture profile does not have the same setting.	Modify the settings in the capture profile or the application plan so that the defined location does not conflict. For example, if you define the application map to store the captured files in a folder, ensure that you save documents to a folder in the capture profile.

I cannot process any additional HyperLearn fonts

Cause	Resolution
The maximum number of fonts (50) have been defined.	To process more fonts, you need to remove at least one of the existing fonts. Doing so disables any application plans using that font. Keeping track of the fonts your application plans use is the best way to avoid this. If you get a message that the font has been processed, the font is stored in Perceptive Content.

Customize item appearance

What is view action?

View action determines how items display in ImageNowExplorer.

When you create a LearnMode or Manual application plan, you configure view action to return documents, records, folders, or record folders based on learned screens from your business application, selected conditions, and user input.

For example, in your school admissions office, your Perceptive Content administrator configures view action to present data captured from the business application that is filtered by document type and student identification number. When used, the view action returns the documents or folders with matching content.

You set the view action using a quick or full view filter, and then test to ensure it returns the current results from your business application. When creating an application plan, you must set a view action for each screen of your business application. View action visually points to the content to display in ImageNowExplorer.

Configure view action for documents

A view action determines how documents display in ImageNowExplorer. To configure view action for documents, complete the following steps.

1. In **Application Plan Designer**, click the **View Action** tab.
2. In the **Screens** pane, select the screen for which you want to configure view action.
3. From the **Type** list, select **Document**.
4. From the **View** list, select the document view you want to associate with this application plan.
5. If you want to modify the document view properties, perform one of the following actions.

Situation	Steps
Quick Filter	• On the **View Action** tab, for each property you want the system to display, click the first column to add the binoculars icon.
Full View	1. On the **View Action** tab, select **Advanced**. 2. Click the **Add** button. 3. In the **Add Condition** dialog box, in the **Constrain By** list, select the kind of constraint that will set up the lists you need in the **Type**, **Field**, and **Operator** boxes. 4. In the **Type** list, click **Normal** or **Variable**. Note If you are creating a condition for a filter, the **Type** list will also include

Situation	Steps
	Prompted. 5. In the **Field** list, click the document field in which the value will be tested by the operator and value you select below. 6. In the **Operator** list, click the operator. 7. In the **Value** box, enter or select the value to which the operator will be applied. Note If you selected **Prompted** in a previous step, the **Message** box replaces the **Value** box. Enter instructions telling the user what value to enter. 8. Click **OK**. 9. Repeat the previous steps for any additional condition rows. **Note:** Each new row begins with **AND** by default. To change **AND** to **OR** or vice versa, click it.

6. Click **Preview** to verify the system returns the current results. Modify column names or column sizes as necessary.

   Perceptive Content displays the changes the next time you execute the View Action.
7. Click **OK**.
8. Close **Application Plan Designer**.

Configure view action for folders

A view action determines how folders display in ImageNowExplorer. To configure view action for folders, complete the following steps.

1. In **Application Plan Designer**, click the **View Action** tab.
2. In the **Screens** pane, select the screen for which you want to configure the view action.
3. From the **Type** list, select **Folder**.
4. From the **View** list, select the folder view you want to associate with this application plan.
5. If you want to modify the folder view properties, perform one of the following actions.

Situation	Steps
Create a Quick Filter display	• On the **View Action** tab, for each property

Situation	Steps
	you want the system to display, click the first column to add the binoculars icon.
Create a Full View display	1. On the **View Action** tab, select **Advanced**. 2. Click the **Add** button. 3. In the **Add Condition** dialog box, in the **Constrain By** list, select the kind of constraint that will set up the lists you need in the **Type**, **Field**, and **Operator** boxes. 4. In the **Type** list, click **Normal** or **Variable**. Note that if you are creating a condition for a filter, the **Type** list will also include **Prompted**. 5. In the **Field** list, click the document field in which the value will be tested by the operator and the value you select below. 6. In the **Operator** list, click the operator. 7. In the **Value** box, enter or select the value to which the operator will be applied. Note that if you selected **Prompted**, the **Message** box replaces the **Value** box. Provide instructions that tell the user what value to enter. 8. Click **OK**. 9. Repeat these steps for any additional condition rows. Note that each new row begins with **AND** by default. To change **AND** to **OR** or vice versa, click it.

6. Click **Preview** to verify the system returns the current results. Modify column names or column sizes as required.

   Perceptive Content displays the changes the next time you execute the **View Action**.
7. Click **OK**.
8. Close the **Application Plan Designer**.

Configure view action for records

A view action determines how records display in ImageNowExplorer. To configure a view action for records, complete the following steps.

1. In **Application Plan Designer**, click the **View Action** tab.
2. In the **Screens** pane, select the screen for which you want to configure a view action.

3. From the **Type** list, select **Record**.
4. From the **View** list, select the record view you want to associate with this application plan.
5. If you want to modify the record view properties, do one of the following.

Situation	Steps
Quick Filter	• On the **View Action** tab, for each property you want the system to match during the search, click the first column to add the binoculars  icon.
Full View	1. On the **View Action** tab, select **Advanced**. 2. Click the **Add**  button. 3. In the **Add Condition** dialog box, in the **Constrain By** list, select the kind of constraint that will set up the lists you need in the **Type**, **Field**, and **Operator** boxes. 4. In the **Type** list, click **Normal**, **Prompted**, **LearnMode** or **Variable**. If you are creating a condition for a filter, the **Type** list will also include **Prompted**. 5. In the **Field** list, click the record field. Perceptive Content tests the value of this field against the value in the **Value** box using the operator you select in the **Operator** list. 6. In the **Operator** list, click the operator. 7. In the **Value** box, enter or select the value to which the operator will be applied. If you selected **Prompted** in step 3, the **Message** box replaces the **Value** box. Enter instructions telling the user what value to enter. 8. Click **OK**. 9. Repeat steps 1-7 for any additional condition rows. **Note:** Each new row begins with **AND** by default. To change **AND** to **OR** or vice versa, click it.

6. Click **Preview** to verify the system returns the current results. Modify column names or column sizes

as necessary.

Perceptive Content displays the changes the next time you execute the View Action.

7. Click **OK**.
8. Close **Form Designer**.

Configure view action for record folders

A view action determines how record folders display in ImageNowExplorer. To configure a view action for record folders, complete the following steps.

1. In **Application Plan Designer**, click the **View Action** tab.
2. In the **Screens** pane, select the screen for which you want to configure a view action.
3. From the **Type** list, select **Record Folder**.
4. From the **View** list, select the record folder view you want to associate with this application plan.
5. If you want to modify the record folder view properties, complete one of the following procedures.

Situation	Steps
Create a Quick Filter display	• On the **View Action** tab, for each property you want the system to match during the search, click the first column to add the binoculars  icon.
Create a Full View display	1. On the **View Action** tab, select Advanced. 2. Click the **Add**  button. 3. In the **Add Condition** dialog box, in the **Constrain By** list, select the kind of constraint that will set up the lists you need in the **Type**, **Field**, and **Operator** boxes. 4. In the **Type** list, click **Normal**, **Prompted**, **LearnMode**, or **Variable**. 5. In the **Field** list, click the record folder field. Perceptive Content tests the value of this field against the value in the **Value** box using the operator you select in the **Operator** list. 6. In the **Operator** list, click the operator. 7. In the **Value** box, enter or select the value to which the operator will be applied. If you selected **Prompted** in the **Type** list, the **Message** box replaces the **Value** box. Enter

Situation	Steps
	instructions telling the user what value to enter. 8. Repeat steps above for any additional condition rows. 9. Click **OK**. **Note:** Each new row begins with `AND` by default. To change `AND` to `OR` or vice versa, click it.

6. Click **Preview** to verify the system returns the current results. Modify column names or column sizes as necessary.

   Perceptive Content displays the changes the next time you execute the **View Action**.
7. Click **OK**.
8. Close **Application Plan Designer**.

Work with application plans

What is an application plan?

An application plan assigns metadata to a captured item, which can be a document or a record, and allows users to easily filter and retrieve that item.

Perceptive Content uses an application plan to assign drawer, folder, and document property values to a captured document. Perceptive Content also uses an application plan to assign file plan, category, folder, and record property values to a captured record.

These values store additional information about your items and allow users to easily filter and retrieve items stored in Perceptive Content.

About selecting an application plan type

Perceptive Content has different types and methods available to capture data from a business application and to create an application plan.

The first step to working with application plans is selecting the type and method that works best with your business environment. The application plan types are Agent, External, External User, Interact, LearnMode, Manual, and Mobile. The Agent type has two methods: Fax Agent and Import Agent. The Interact type has two methods: Connector for SAP ArchiveLink and Interact for Outlook. The LearnMode type has six methods: DOS command prompt, HyperLearn, Internet Explorer, Terminal Capture, Viewpoint, and Window Walker.

Finding the right type and method is sometimes a case of trial and error and involves finding the right method that fits the specific needs of your business. When using a business application, you need to understand how Perceptive Content captures data from the application, its design, and the property values you need to capture to link your images. It is extremely helpful to understand what type of business application you use, the type of interface it uses, and how it is accessed by your users.

Application plan types

When creating an application plan, select the type that works best for your business environment.

Agent

Use this application plan type to map data to content captured by Fax Agent or Import Agent.

External

Use this application plan type to map data to documents using iScript or another external Perceptive Content interface. You cannot capture records with an External application plan.

Interact

Use this application plan type to map data to content captured by an Interact client, such as Interact for Lexmark.

LearnMode

Use this application plan type to map data to content gathered from one or more screens in your GUI-based, DOS-based, Web-based, or terminal emulation business application.

Manual

Use this application plan type to manually map data to content.

Mobile

Use this application plan type to map data to a document that is uploaded from a mobile device. You cannot capture records with a Mobile application plan.

External User

Use this application plan type to map data to a document using an external Perceptive Content interface.

Create an application plan

To create an application plan, complete the following steps.

The Mobile application plan type does not support Records Manager functionality. To view records functionality, you must install a Records Manager license.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.

2. In the left pane, expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select **New**.
4. In the **Plan Settings** dialog box, select the **General** tab.
5. Under **Information**, complete the following substeps.
    1. For **Content Type**, select the **Document** or **Record** option. You cannot change the content type once the application plan is created.
    2. In the **Name** box, type a unique name for your application plan.
    3. Optional. In the **Description** box, type a description for your application plan.
    4. Optional. Clear the **Is active** check box if you do not want this application plan to be active.
6. Click **OK**.

Create an application plan overview

To create a new application plan, complete the following sequence of procedures.

1. Create an application plan.
2. Add screens to your application plan.
    - What is a screen?
    - Add a screen.
3. Define dictionary data elements for your application plan.
    - What is the dictionary?
    - Add a data element to the dictionary.
    - Define a data dictionary element for a DOS program.
    - Define a dictionary element using Internet Explorer.
    - Define a data dictionary element using TerminalCapture.
    - Define a data dictionary element using Viewpoint.
    - Define a data dictionary element using HyperLearn.
    - Define a data dictionary element using Win Walk.
4. Create a document, folder, record, record folder, or shortcut map.
    - Create an application map.
5. Create application plan map views.
    - What is view action?
    - Configure view action for documents.
    - Configure view action for folders.
    - Configure view action for records
    - Configure view action for record folders
6. Apply security to the application plan.
    - Add a user or group to an application plan.
    - Remove a user or group from an application plan.

Activate an application plan

An application plan assigns metadata to a captured item, which can be a document or a record, and allows users to easily filter and retrieve that item. The following steps describe how to activate an application plan.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application you want to activate and click **Settings**.
4. In the **Plan Settings** dialog box, select**Is active**.
5. To close the **Plan Settings** dialog box, click **OK**.

Modify an application plan

To modify an application plan, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application plan you want to modify.
4. To modify an application plan, perform any of the following actions.

Situation	Steps
Modify an application plan	1. Click **Modify**. 2. In **Application Plan Designer**, make the necessary changes to the application plan, such as modifying maps, or adding new screens or dictionary data elements. If your application plan has more than one screen, you must modify each screen separately. 3. Click **Save** when you are finished modifying the application plan and close **Application Plan Designer**.
Modify application plan settings	1. Click **Settings**. 2. In the **Plan Settings** dialog box, make the necessary changes to the application plan's general settings, such as changing the application plan name and description, or updating security preferences. 3. Click **OK**.

Copy an application plan

To copy an application plan, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application plan you want to copy.
4. Click **Copy**.
5. In the **Copy** dialog, enter a name and click **OK**.
6. To modify the copied version of the application plan, complete any of the following actions.
   - To assign a name to the copied application plan, click **Rename**, type the new name, and press ENTER.
   - To make changes to the application plan, such as modifying context maps or adding new screens or dictionary data elements, select the copied version and click **Modify**.
   - To make changes to the general settings of the application plan, such as changing the application plan name or description, or to make changes to the security settings, select the copied version and click **Settings**.

Rename an application plan

To rename an application plan complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application plan you want to rename.
4. Click **Rename**.
5. Type the new application plan name and press ENTER.

Delete an application plan

To delete an application plan, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application plan you want to delete.
4. Click **Delete**.
5. In the confirmation dialog box, click **Yes**.

Agent

About using the Agent type methods

You use the Agent type application plan with Import Agent and Fax Agent to capture data directly into Perceptive Content.

- Fax Agent. The Fax Agent enables you to capture incoming faxes directly into Perceptive Content by converting them into documents. This method uses data available from the Fax Agent to identify possible document property values. Fax-specific data includes the fax sender number, fax sender DID, and incoming fax channel.
- Import Agent. The Import Agent, installed with the ImageNow Server, handles all automated import routines on the server. This method uses data available from the Import Agent to identify possible item property values. Import specific data includes Import mode, Import file name, Import file location, Import index file line, and Import associated text file.

Create a Fax Agent application plan

To create a Fax Agent application plan, complete the following sequence of procedures.

1. Create an application plan.
2. Define Fax Agent dictionary values.
3. Optional. Add a macro.
4. Create an application plan map.
5. Add a user or group to an application plan.

Create an Import Agent application plan

To create an Import Agent application plan, complete the following steps.

1. Create an application plan.
2. Define Import Agent dictionary values.
3. Optional. Add a macro.
4. Create an application plan document map.
5. Add a user or group to an application plan.

External

About using the External type

You use the External type application plan with external Perceptive Content interfaces, such as iScript and Integration Server.

This type uses data available from the external interface to identify possible property values for items and containers. It also limits the property values you can use for input in the external interface.

Configure an External application plan

You use the External application plan to map data to items using iScript or another external Perceptive Content interface. To create an external type application plan, complete the following steps.

1. Create an application plan.
2. Add a data element to the dictionary.
3. Add a macro.
4. Create an application plan map.
5. Add a user or group an application plan.

Create an application plan

To create an application plan, complete the following steps.

The Mobile application plan type does not support Records Manager functionality. To view records functionality, you must install a Records Manager license.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select **New**.
4. In the **Plan Settings** dialog box, select the **General** tab.
5. Under **Information**, complete the following substeps.
    1. For **Content Type**, select the **Document** or **Record** option. You cannot change the content type once the application plan is created.
    2. In the **Name** box, type a unique name for your application plan.
    3. Optional. In the **Description** box, type a description for your application plan.
    4. Optional. Clear the **Is active** check box if you do not want this application plan to be active.
6. Click **OK**.

Add a data element to the dictionary

A data element is a piece of information collected from your associated business application screen. To add a data element to the dictionary, complete the following steps.

1. In the **Dictionary** pane, click the **New Field** button.
2. In the **Dictionary Field** dialog box, under **General**, complete the following actions.
    1. In the **Name** box, type a unique name for your new data element.
    2. Under **Processing**, add a macro or a script.
3. Optional. In the **Dictionary Field** dialog box, under **Sample**, complete one of the following actions.
    - If you selected a macro as your processing type, type an example of the unaltered data in the **Input** box.

- If you selected a script as your processing type, type an example of the unaltered data in the **Input** box. Click **Refresh** to update the results and verify the script returns the correct data.

4. Click **OK** to save your changes.

Add a macro

A macro is a script you add to an application plan to define how Perceptive Content formats and processes data elements before the elements are added to the dictionary. To add a macro, complete any of the following procedure.

- Add an alphanumeric macro.
- Add a concatenation macro
- Add a date conversion macro
- Add a delete characters macro
- Add a fixed length macro
- Add a line search macro
- Add a number format macro
- Add a split attribute at delimiter macro
- Add a static search macro
- Add a trim characters macro

Add an alphanumeric macro

To add a macro that handles alphanumeric characters for a dictionary field in an application plan, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Alpha numeric macro** and click **OK**.
4. In the **Alpha Numeric**dialog box, under **General**, select one of the following options from the **Alphanumeric Filter**list.
   - Select **Alphanumeric Characters Only** if you want the system to remove non-alphanumeric characters from the output string. These characters include all symbols and punctuations.
   - Select **Alpha Characters Only**if you want the system to remove non-alphabetic characters from the output string.
   - Select **Numeric Characters Only** if you want the system to remove non-numeric characters from the output string.
5. Click **OK** until you return to .

Add a concatenation macro

To add a macro that concatenates characters in a dictionary field for an application plan, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.

2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Concatenation** and then click **OK**.
4. In the **Concatenation** dialog box, under **General**, click the ellipsis button in the **Concatenation Rule** field.
5. In the **Concatenation Rule Builder** dialog box, under **General**, double-click the dictionary data elements you want to join together.
6. Click **OK** until you return to **Application Plan Designer**.

Add a date conversion macro

To create a macro that converts values from your business application into a format that document, folder, and shortcut properties can accept, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Date Conversion**and click **OK**.
4. In the **Date Conversion** dialog box, under **General**, select the appropriate format from the **Format** list or enter a unique format.

   Note: The date format you select or enter must match the date format of the data element linked to the selected dictionary item. If you want to assign a value to a date custom property, you need to select this macro option to convert the date from your business application into the format of year, month, and day.

5. Click **OK**.

Add a delete characters macro

To add a macro that deletes unnecessary characters from your business application values, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Delete Characters** and then click **OK**.
4. In the **Delete Characters** dialog box, under **General**, type the characters you want the system to remove in the **Delete Characters**box.
5. Click **OK**.

Add a fixed length macro

To add a macro that defines the number of characters a value from your business application can have, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro**dialog box, select **Fixed Length** and then click **OK**.

4. In the **Fixed Lengths**dialog box, under **General**, perform the following substeps:
    1. From the **Length** list, enter the number of characters or digits you want the value to have.
    2. From the **Direction** list, select the direction from which you want the system to cut off the characters.
5. Click **OK**.

Add a line search macro

To create a macro that captures values from your business application based on a line on the screen, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Line Search** and then click **OK**.
4. In the **Line Search** dialog box, in the **Line Count** box, type the line from which you want the system to capture the data.
5. From the **Direction** list, select the direction from which you want the system to search.
6. In the**Start Character Position** box, type the number of the starting character position.
7. From the **Stop Method** list, do one of the following actions:
    - Select **End of Line** if you want the stop method to be determined by the end of a line.
    - Select **Character Length** if you want the stop method to be determined by a fixed number of characters and then enter the amount of characters in the **Length** box.
    - Select **Stopping Characters** if you want the stop method to be determined by a specific string of characters and then type those characters in the **Characters**box.

    Note: You may need to expand the **Stop Method** tree to view the **Length** and **Characters** boxes.
8. Click **OK**.

Add a number format macro

To add a macro that formats numeric values from your business application, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Number Format** and then click **OK**.
4. In the **Number Format** dialog box, under **General**, complete the following substeps.
    1. From the **Decimal** symbol, select the value you want to represent the decimal system.
    1. In the **Decimal places** box, enter the number of digits you want the system to display after the decimal place.
    2. Click **OK**.

Add a split at delimiter macro

To create a macro that defines the character that separates the values in a string in your business application, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro**dialog box, select **Split at Delimiter** and then click **OK**.
4. In the **Split at Delimiter** dialog box, under **General**, complete the following substeps,
   1. In the **Delimiter** box, type a character that separates the values in a string that you want to search.

      The system captures the value between the character you specify to the next delimiter.
   2. In the **Position in string** box, type a number that indicates the location of the delimiter to the right of the value that you want to find in the string.

   Note: You can specify any delimiter character such as a space, carat, dash, underscore, or slash. You can also specify multiple characters separated by commas.
5. Click **OK**.

Add a static search macro

To add a macro that captures data to the right or left of the static string to the end or beginning of the line in a business application, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro**dialog box, select **Static Search** and then click **OK**.
4. In the **Static Search** dialog box, in the **Anchor Text** box, type the value of the static string.

   The system captures data to the right or left of the static string to the end or beginning of the line.
5. From the **Line Offset** list, perform one of the following actions.
   - Select **Yes** if the line number of the static string value can change.
   - Select **No** if the line number of the static string does not change.

   Example The static string may move down a row if a header is added to the screen.
6. From the**Line Location**list, complete one of the following actions.
   - Select **Above Anchor** if you want the system to return the value of the line above the text you specified in the **Anchor Text**box.
   - Select **Below Anchor** if you want the system to return the value of the line below the text you specified in the **Anchor Text**box.
7. In the **Line Count** box, type the line from which you want the system to capture the data.
8. In the **Direction** list, select the direction you want the system to start its search.
9. From the**Stop Method** list, complete one of the following actions.
   - Select **End of Line** if you want to end the system search at the end of the current line.

- Select **Character Length** to end the system search after a fixed number of characters from the starting position and enter the amount of characters in the **Length** box. Note You may need to expand the **Stop Method** tree to view the Length box.
- Select **Stopping Characters**to specify a stopping character or a group of stopping characters and type those characters in the **Characters** box. Note You may need to expand the Stop Method tree to view the **Characters** box.

Note: You may need to expand the **Stop Method** tree to view the **Length** and **Character** boxes.

10. Click **OK**.

Add a trim characters macro

To add a macro that trims characters from a string value in a business application, complete the following steps.

1. In the **Dictionary** pane, double-click the data field where you want to apply a macro.
2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Macro**.
3. In the **Select Macro** dialog box, select **Trim Characters** and then click **OK**.
4. In the**Trim Characters** dialog box, under **General**, complete the following substeps.
    1. In the **Line Number**box, type the line from which you want the system to capture the data.
    2. From the **Direction** list, select the direction from which you want the system to trim off the characters.
    3. In the **Trim Characters** box, enter the characters you want the system to trim.
5. Click **OK**.

Create an application plan document map

To create an application plan document map, complete the following steps.

1. Define an application plan map drawer.
2. Define an application map folder level. You may define more than one folder level.
3. Define an application plan map document level.
4. Optional. Define an application map shortcut.
5. Define an application map drawer.
6. Test an application plan map.

Add a user or group to an application plan

Adding a user or group to an application plan allows those users to access and utilize it. To add a user or group to an application plan, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.

3. In the right pane, on the **Applications** tab, select the application to which you want to add a user or group and click **Settings**.
4. In the **Plan Settings** dialog box, on the **Security** tab, click **Add**.
5. Under **Privileges**, grant or deny the specific privilege actions.
   - To grant specific application plan privileges for the selected user or group, click the column in front of each privilege until  appears.
   - To deny specific application plan privileges for the selected user or group, click the column in front of each privilege until  appears.

   Note: You can grant or deny privileges to create or link documents, automatically create folders when the document is linked, run the active screen's view action and manage the application plan. Alternatively, you can click Allow All, Deny All, or Deselect All to allow, deny or deselect all privileges for the selected user or group.
6. Click **OK** to close the **Plan Settings** dialog box.

External User

About using the External User type

You use the External User application plan when capturing files or scanning images from an application that is utilizing Integration Server to connect with Perceptive Content, such as Perceptive Experience Content Capture.

When using an External User type plan, the user can enter some of the properties, and you can configure which fields the user is able to modify. You can also create dictionary data elements when using this type of application plan.

Configure an External User application plan

You use the External User application plan when capturing files or scanning images from an application that is utilizing Integration Server to connect with Perceptive Content. To configure an External User application plan, complete the following steps.

1. Create an application plan.
2. Add a data element to the dictionary.
3. Add a macro.
4. Create an application plan document map.
5. Add a user or group to an application plan.

Interact

About using the Interact type methods

You use the Interact type to integrate the management features of Perceptive Content with your existing SAP and Outlook applications.

The Interact type contains two methods, the Connector for SAP ArchiveLinkand Interact for Outlook. Linking items according to content guarantees long-term easy access to the items.

The following bullets describe each type.

- Connector for SAP ArchiveLink. This integration allows you to deliver secure, digital versions of paper and other electronic documents associated with SAP transactions. SAP ArchiveLink is a service integrated in the SAP Web Application Server for linking archived documents and the application documents entered in the SAP system.
- Interact for Outlook. This integration uses a capture profile to determine how to link single or multiple captured emails and add them to Interact for Outlook.

Create an Interact application plan

You use the Interact application plan to map data to items captured by an Interact client, such as Interact for Lexmark. To create an Interact application plan, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans** and select **Interact**.
3. On the **Applications** tab, select **New**.
4. In the **Plan Settings** dialog box, select the **General** tab.
5. Under **Information**, complete the following steps.
    1. For **Content Type**, select the **Document** or **Record** option.

        Note: You cannot change the content type once the application plan is created.

    2. In the **Name** box, type a unique name for your application plan.
    3. Optional. In the **Description** box, type a description for your application plan.
    4. Clear the **Is active** check box if you do not want this application plan to be active.
    5. Click **OK**.

LearnMode

LearnMode type methods

The LearnMode type application plan uses methods to learn screens from your business application and links data to your application plan. There are six different methods to learn your business application, and the best method to use depends on the type of application you are using. For example, knowing whether your application is GUI-based, web-based, DOS-based, a terminal emulator, or other type of application will assist in determining the LearnMode type to use.

GUI-based methods

- Window Walker
- HyperLearn
- Viewpoint

Other methods

- DOS Command Prompt
- Internet Explorer
- Terminal Capture

What is the DOS Command Prompt method?

The DOS Command Prompt method is the LearnMode type method most often used with DOS-based programs.

These programs behave very similarly to programs running under terminal emulation. Therefore, this method is very similar to the Terminal Capture method. LearnMode captures the screen data and copies it to the system clipboard. The clipboard content is then copied into the captured window in the screen. Properties are defined by highlighting the text in the captured window. The screen label is defined in the same manner.

The most common type of program that uses a DOS window is a legacy program written prior to the widespread use of the Windows graphical user interface. If you are uncertain whether your business application runs in a DOS window, contact your Administrator.

What is the Internet Explorer method?

The Internet Explorer method is the LearnMode type method you use with web-based applications.

Business applications that are rendered through HTML, ActiveX, Java, Macromedia Flash, or some combination of these are easily integrated with Perceptive Content.

LearnMode using the Internet Explorer method utilizes an HTML parser that conducts browser interrogation to identify property values. Data is obtained automatically and dynamically from the web application running on your desktop. When capturing data from your business application, LearnMode delivers the unique controls to the application plan. These controls are used in the captured window to define document property values and the screen label.

What is the Terminal Capture method?

The Terminal Capture method is the LearnMode type method you use with terminal emulator programs and mainframe programs.

This method uses a straightforward technique of copying the business application screen contents to the system clipboard. This method is most commonly used with Terminal Emulation programs. The terminal emulator displays data screens from a mainframe or online business application.

Depending on the terminal emulator that you use, the way it copies screen data to the clipboard varies. A connect string is needed for each screen that is created. The connect string provides LearnMode with a set of system level program calls that relate specifically to the terminal emulation program being used. Determining the connect string is one of the most difficult tasks in this method. The clipboard contents are then copied into the captured window in the screen. Document property values are defined by highlighting the text in the captured window. The screen label is defined the same way.

We recommend checking with Perceptive Software Product Support to ask if the connect strings have already been determined for the terminal emulator for another customer using the same program.

What is the Viewpoint method?

The Viewpoint method is the LearnMode type method that is a hybrid between the Window Walker and HyperLearn methods and is used with some GUI applications on the Windows platform.

It is similar to Window Walker in that it is control-based. However, instead of using address of node in the control tree to identify and locate data, Viewpoint (formerly known as Window Text) uses x,y coordinates (similar to HyperLearn) to locate a control. The controls can be in a disabled state and the window can be dragged off the screen. Property values are defined by clicking the pointer in each text box in the captured window. This method also usually defines the screen label for you by collecting the text in the title bar of the captured window.

Perceptive Content includes an enhancement to LearnMode using the Viewpoint method to expand options for learning web-based applications that were built with tools such as Macromedia Flash and for use by Mozilla Firefox users.

What is the Window Walker method?

The Window Walker method is the LearnMode type method you use with some GUI applications that conform to Microsoft Windows interface design guidelines.

Many applications in use today have a graphical user interface that operates entirely within a single window on the desktop. Examples of applications that run inside a single window are Microsoft Word and Microsoft Excel. Custom-built applications also may fall into this category. Additionally, Window Walker can learn some Java applets and applications that run in browsers.

These application windows are published to the operating system in such a way that LearnMode can easily communicate with them using the Window Walker method. Different information is contained within easily distinguishable controls, such as fields, images, and objects. Data is obtained automatically and dynamically

from the Windows application running on your desktop. When capturing data from your business application, Window Walker method it delivers the unique control IDs to the application plan for defining property values and the screen label.

Create a Learn Mode application plan

You use the LearnMode application plan to map data, gathered from one or more screens, to documents or records where each screen represents a learned screen or window in your business application. To create a LearnMode application plan, complete the following steps.

Prerequisite Start your business application, then open the window and active business object that you want Perceptive Content to learn.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans**, and select **LearnMode**.
3. In the right pane, on the **Applications** tab, select **New**.
4. In the **Plan Settings** dialog box, select the **General** tab.
5. Under **Information**, complete the following substeps.
    1. Next to **Content Type**, select either **Document** or **Record**.

        Note: You cannot change the content type once the application plan is created.

    2. In the **Name** box, type a unique name for your application plan.
    3. Optional. In the **Description** box, type a description for your application plan.
    4. Clear the **Is active** check box if you do not want this application plan to be active.
6. Under **Application**, select the method you want to use to identify the business application.

    Note:

    - The methods used to identify your business application allow you to use wildcard symbols.
    - The window title in **LearnMode** is limited to 128 characters.
7. Click **OK**.

    Perceptive Content captures a screen shot of the business application window and displays it in **LearnModeApplication Plan Designer**.

Identify a window

To identify a window title in a LearnMode application plan, complete the following steps.

Prerequisite Before you create or modify a LearnMode application plan, start your business application, then open the window and active business object that you want Perceptive Content to learn.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans**, and select **LearnMode**.

3. On the **Applications** tab, select the appropriate application plan and click **Modify**.
4. In the **Plan Settings** dialog box, from the **Method** list, select **Window Title**, **Class ID**, or **Application Name**, then perform one of the following actions.
    1. If you want **Perceptive Content** to identify the application or screen, click **Identify**.
    2. In the **Window Select or Starter** dialog box, click **Start**.
    3. Point to the title bar of the business application window and click the mouse button. The data captured from the business application window appears in the **Value** box.
    4. Click **Test**.

       The system attempts to find the application window or screen with the given string in the **Value** box.
5. Click **OK**.

   Perceptive Content captures a screen shot of the business application window and displays it in **Application Plan Designer**.

Define a dictionary data element for a DOS program

A data element is a piece of information collected from your associated business application screen. To define a dictionary data element using a DOS method, complete the following steps.

Prerequisite Before you add a data element to the dictionary, you must create a screen. You also need to ensure you have your business application open to the screen and business object you want to capture.

1. In **Application Plan Designer**, in the **Elements** tab, select a **DOS Command Prompt**from the **Method** list.

   The system copies the screen contents from the business application and displays it in the **Application Plan Designer**.
2. To add a data element, perform one of the following actions.
    - If the data element is already added to the dictionary, use the mouse to highlight the data in the captured window that you want to define, and then drag it to the appropriate data element in the **Dictionary** pane.
    - If the data element is not in the dictionary, use the mouse to highlight the data in the captured window that you want to define, and then drag it to the bottom of the data element list in the **Dictionary** pane.
3. In the **Dictionary Field** dialog box, add the data element as you normally would.
4. Repeat these steps for each data element you want to define.

HyperLearn Settings

What is the HyperLearn method?

The HyperLearn method is the LearnMode type method used with GUI applications that are not as accessible to Perceptive Content as others.

Some custom-built applications, for example, may not adhere to all current Windows interface standards, and can only deliver bitmapped data about a Windows screen. In this case, LearnMode using the HyperLearn method utilizes an on-screen character recognition process to gather data from the business application. Basically, the HyperLearn method takes a screen shot of the business application window, analyzes its content, and converts the screen characters into text.

In other methods, field data is gained from controls. HyperLearn instead intelligently interprets the characters presented within a defined region on the window, including the title bar. Properties are defined by drawing boxes around the content you want in the captured window.

Using this method requires that you know the specific window font information, type face, point size and formatting style, used by the business application. You may need to contact the original manufacturer or designer of the business application to obtain font information.

What are HyperLearn fonts?

Perceptive Content can store up to 50 screen font files for your LearnMode application plans that use the HyperLearn method.

The LearnModetype HyperLearn method works using the specific business application window font information, type face, point size and formatting style. When using the HyperLearn method to create a screen linked with your business application, the font is used by Perceptive Content for character recognition in the business application window. Each LearnMode type application plan using the HyperLearn method can use one or more of the stored font files.

Example fonts for HyperLearn

Creating a screen in a LearnMode application plan, using the HyperLearn type, requires that you know the specific screen font, font style, and font size used by the business application.

In working with HyperLearn, the following fonts are typically used in most business applications. Perceptive Content stores the font in the *[drive:]\Program Files\ImageNow\hyperlearn* folder on the ImageNow Server. The font size normally used in most business applications varies between 8 to 10 points, but some business applications use 12 point.

Font	Style	Size
Arial	Regular	8
Arial	**Bold**	**8**
Arial	Regular	10
Arial	**Bold**	**10**
Courier	Regular	8
Courier	**Bold**	**8**
Courier	Regular	10
Courier	**Regular**	**10**
MS Sans Serif	Regular	8
MS Sans Serif	**Bold**	**8**
MS Sans Serif	Regular	10
MS Sans Serif	**Bold**	**10**
Tahoma	Regular	8
Tahoma	**Bold**	**8**
Tahoma	Regular	10
Tahoma	**Bold**	**10**
Times New Roman	Regular	8
Times New Roman	**Bold**	**8**
Times New Roman	Regular	10
Times New Roman	**Bold**	**10**

Set or modify the fonts for a business application screen

When using HyperLearn for creating or modifying a screen in a LearnModeapplication plan, you can select different fonts to use in your business application screen. To set or modify fonts for your business application screen, complete the following steps.

Prerequisite Start your business application and open the window that you want LearnModeto learn, and navigate to an active business object.

1. In **LearnMode** , in the **Screens** pane, select the appropriate screen.
2. On the **Screen Elements** tab, click **Capture**.
3. After your business application window or screen is captured, click **Settings**.
4. In the **HyperLearn Settings** dialog box, add and remove the **Fonts** list as needed, and then click **OK**.
5. Optional. Test the item property definitions in the application plan with the new font settings.

Next When you modify an existing template, you must re-capture the window or screen of your business application to use this procedure.

Set HyperLearn settings

You use this LearnMode method for GUI applications that do not make themselves as accessible to Perceptive Content as others. To set HyperLearn settings, complete the following steps.

1. On the **Perceptive Content** toolbar, click **Settings > Options**.
2. In the **ImageNowOptions** dialog box, in the left pane, click **LearnMode**, and then perform any of the following optional actions:

Situation	Step
To set the delay time for screen captures	• In the **Delay** list, select the time.
To display an alert when Perceptive Content encounters an unknown character when reading a business application screen	• Select the **Display message when unknown characters are found** check box.
To restrict font templates to the local computer	• Select the **Use font templates from local machine only** check box.
To convert an unwanted color appearing in the document key or screen ID clip region	1. Select the **Enable color conversion** check box. 2. In the **Search for** box, specify the unwanted color. 3. In the **Replace** box, specify the color to convert to.
To allow dynamic relocation of the target area	1. Select the **Enable position offset for dynamic fields** check box. 2. Set pixel movement to shift the search area when a business application's fields shift. 3. In the **Shift Region up/down** box, specify the pixel movement up or down. 4. In the **Shift Region left/right** box, specify the pixel movement left or right.
To troubleshoot issues with application plans by providing screen capture output	1. Select the **Enable debug** check box. 2. To view the output on the screen, select **Screen**. 3. To produce an output file, select **File (hyperlearn.log)**.

Note: Enabling position offset for dynamic fields shifts the coordinates for all HyperLearn application

plans and templates on the client computer. It is possible this setting can disable other templates. However, because the screen shifts the same direction on the same business application every time, this is normally not an issue.

3. Click **OK**.

Terminal Emulator Settings

What are connect strings?

Connect strings contain commands to gather data from a terminal emulation program screen with the LearnMode type Terminal Capture method.

Complete connect strings have two basic parts. The first part is a set of connect strings that issue the "Select All" command. Most programs have this type of command under the Edit menu. Within your terminal emulation program, locate commands such as Select All, Select Screen, or Select Display. The second part is a set of connect strings that copies the selected data to the Clipboard. Most terminal emulation programs contain a command to copy data.

This LearnMode type uses a copy and paste technique to copy data from the current terminal emulation program screen to the Clipboard. From the Clipboard, the data is captured by the LearnMode application plan. Perceptive Content can send commands in the form of connect strings, to the terminal emulation program to instruct it to perform certain tasks. Connect strings are made up of sets of numbers and other characters that are interpreted as system or keyboard commands. For example, a complete set of connect strings is "ms;7022;" for the Tiny Term Plus terminal emulator. Essentially, if you can copy the contents of your business application screen using keys on your keyboard or the mouse, you should be able to use connect strings and the LearnMode Terminal Capture method.

Virtual Terminal Emulator commands

This topic lists extended descriptions, parameters, defaults, and examples for each virtual terminal emulator command.

About keyboard commands

Several virtual terminal emulator commands simulate keyboard commands to help you create connect strings for the LearnModetype, Terminal Capture. These commands can assist in compiling a complete set of connect strings when you cannot use the Spy++ utility to obtain one or both commands that are necessary to select the screen contents or copy the screen contents. You can also use these commands in conjunction with the commands obtained from Spy++ to create connect strings when necessary.

Virtual mouse emulation commands

There are several commands used to emulate mouse movements that can be used as a connect string section. These virtual commands can be used with or without parameters. For example, `ms;` simulates a select mouse action. Use the parameters for `ms` only when you need to specify x,y coordinates or use the right mouse button. An example of the `ms` command with full parameters is `ms(10,10,-10,-10,1,0,150,150,1);` this could be used as a connect string section.

mc (mouse click)

Simulates a mouse click at an x/y offset from the upper left of the terminal window. Used to deselect terminals that do not unselect after a select all/copy. This is the old method. Use the mu command instead whenever possible.

Parameters

x offset from upper left

y off offset from upper left

Parameter defaults

x offset from upper left: 0

y off offset from upper left: 150

Example

```
mc(200,200);
```

md (mouse drag)

Simulates a select all mouse action. This is the old method. Use the ms command instead whenever possible.

Parameters

x offset from upper left

y off offset from upper left

x offset from lower right

y offset from lower right

Parameter defaults

x offset from upper left: 0

y off offset from upper left: 0

x offset from lower right: -1

y offset from lower right: -1

Example

```
md(10,10, -10,-10);
```

ml (mouse lock)

Moves the mouse to 0,0 and prevents the user from moving it until the connect string is executed. This command has no parameters.

Example

```
ml;
```

ms (mouse select)

Parameters

x offset from upper left

y offset from upper left

x offset from lower right

y offset from lower right

drag from top

child window x offset

child window y offset

use right mouse button

Parameter defaults

x offset from upper left: 0

y offset from upper left: 0

x offset from lower right: -1

y offset from lower right: -1

drag from top: 0

child window x offset: 150

child window y offset: 150

use right mouse button: 0

Example

```
ms(10,10,-10,-10,1,0,150,150,1);
```

Additional information

By default, the mouse drags from lower right to upper left - to change this, set the drag from top to 1 instead of 0. By default, the last parameter is set to 0. Set it to 1 to use right mouse button.

mu (mouse unselect)

Simulates a mouse click at an x/y offset from the upper left of the terminal window.

Parameters

child window x offset

child window y offset

x offset from upper left

y offset from upper left

use right mouse button

Parameter defaults

child window x offset: 150

child window y offset: 150

x offset from upper left: 100

y offset from upper left: 100

use right mouse button: 0

Example

```
mu(150,150,70,70,0);
```

Additional information

Used to deselect terminals that do not unselect after a select all/copy. Gets the child window from the offset of the main window upper left and simulates a mouse click at the offset from the upper left of the child.

msx (mouse select)

Simulates a select all mouse action to integrate IDX applications or if the ms command is not successful. Gets the child window from the offset of the main window upper left.

Parameters

x offset from upper left

y offset from upper left

x offset from lower right

y offset from lower right

drag from top

child window x offset

child window y offset

use right mouse button

Parameter defaults

x offset from upper left: 0

y offset from upper left: 0

x offset from lower right: -1

y offset from lower right: -1

drag from top: 0

child window x offset: 150

child window y offset: 150

use right mouse button: 0

Example

```
msx(2,2,-2,0,150,150,0);
```

Additional information

By default, the mouse drags from lower right to upper left. To change this, set the drag from top to 1 instead of 0. By default, the last parameter is set to 0. Set it to 1 to use the right mouse button.

mx (mouse select)

Another way to select an application screen. Gets the child window from the offset of the main window upper left.

Parameters

x offset from upper left

y offset from upper left

x offset from lower right

y offset from lower right

drag from top

child window x offset

child window y offset

Parameter defaults

x offset from upper left: 0

y offset from upper left: 0

x offset from lower right: -1

y offset from lower right: -1

drag from top: 0

child window x offset: 150

child window y offset: 150

Example

```
mx(5,5,-5,-5,0,150,150);
```

Additional information

By default, the mouse drags from lower right to upper left. To change the direction, set drag from top to 1 instead of 0.

Virtual miscellaneous emulation commands

There are several miscellaneous commands that can be used as a connect string section.

bl (bottom lines)

For terminals that have a scrolling screen buffer. This performs a copy and then ignores all lines except the number of bottom lines specified in the parameter.

Parameters

number of bottom lines to use

Parameter defaults

number of bottom lines to use: 23

Example

```
bl(25);
```

lf

Use when a terminal emulator ends lines with lf only and the screen can't be captured correctly. Use to convert lf (linefeed) to cr/lf (carriage return, linefeed) before processing the screen data.

Example

```
If;
```

sl (sleep)

Delays for specified number of milliseconds: 1000 = 1 second.

Parameters

number of milliseconds

Parameter defaults

number of milliseconds: 100

Example

```
sl(1000);
```

Virtual keyboard emulation commands

You can use virtual commands to simulate keystrokes in a connect string section. The `kd` command corresponds to the pressing down of a key. The `ku` command corresponds to the release of the key. You can use the `kd` and `ku` commands with any numeric value from the second table in this topic.

kd (key down)

Simulates a key press.

Parameters

Valid numeric value

Example

```
kd65;
```

ku (key up)

Simulates a key release.

Parameters

Valid numeric value

Example

```
ku65;
```

For example, the keyboard shortcut for the Select All command in many programs is CTRL+A. The numeric value for the CTRL key is 17, and the value for A is 65. Thus, the virtual commands for CTRL+A are `KD17; KD65; KU65; KU17;`

The first command, `KD17;` simulates holding down the CTRL key. The next commands, `KD65;` and `KU65;`, simulate pressing and releasing the A key. The last command, `KU17;` simulates releasing the CTRL key. Use this command string section to select all the screen contents of the business application screen in some terminal emulators.

Valid numeric values for keyboard commands

The following table lists the numeric values you can use with the `kd` and `ku` virtual keyboard emulation commands.

Numeric Value	Simulates Command
1	Left Mouse Click
2	Right Mouse Click
3	CTRL+C
8	BACKSPACE
9	TAB
12	Keypad 5 (Num Lock Off)
13	ENTER
16	SHIFT
17	CTRL
18	ALT
19	PAUSE

Numeric Value	Simulates Command
20	CAPS LOCK
27	ESC
32	Spacebar
33	PAGE UP
34	PAGE DOWN
35	END
36	HOME
37	LEFT ARROW
38	UP ARROW
39	RIGHT ARROW
40	DOWN ARROW
44	PRINT SCREEN
45	INSERT
46	DELETE
48-57	ASCII "0" ASCII "9"
65-90	ASCII "A" ASCII "Z"
97-122	a-z

Example Terminal Emulator connect strings

If you have one of the terminal emulators listed below, you can try the connect strings listed in the table for it. If it works, it can save you some time. However, there is no guarantee it will work due to differences in terminal emulator versions and computer systems. For some terminal emulators, you can use Spy++ to verify that the connect strings listed in the table below are the same for your business application.

Emulator	Connect String	Comments
Banker	ls;600;	
CRT	32825;57634;	
Easycomm	kd118;ku118;	This emulator has no menu commands. Pressing the F7 key copies all on-screen data.
EWAN	24321;	
Extra! PC v6.5	33602;57634;	Change settings for edit to copy on a line-by-line basis.
Host Access v7.1	ms;55013;	
HostExplorer	4028;4022;	
NetTerm	?57642;57634;	
?OmniPath	33207;33202;	
QWS3270	1031;1030;	
Rumba AS/400 Display	kd17;kd65;ku65;kd67;ku67;ku17;	Virtual key command string to perform keyboard CTRL+A and CTRL+C (Select All and Copy).
Reflections v6.0 for UNIX and VMS	53;46;	Reflection buffers several hundreds of lines. Must limit the number of lines from the bottom of the screen to 23. Change imagenow.ini file ([LearnMode Prefs]/BottomLines=23).
SmarTerm	32780;57634;kd17;kd81;ku81;ku17;	
TCP3270	4028;4022; 3015;3010;3023;	
Telnet	126;121;	
Tiny Term Plus	ms;7022;	

Emulator	Connect String	Comments
Tiny Term Plus v4.42.2429	ms;7022;If;	ms; uses the mouse to select the screen contents. 7022; copies the contents. If; If to cr/If which is necessary for this version of Tiny Term Plus.
TN3270	57642;57634;c	

Manual

About using the Manual type

You use the Manual type application plan as a stand-alone system, or when a business application is not available.

When using a Manual type application plan, some of the properties can be manually entered. This is different from the LearnMode application plan type where the properties are defined dynamically using values from a captured screen in the business application. There are features available during property definition that can make manual entry less manual.

> For example, predefined lists applied to properties present the user with a list of choices when linking an image, and scripts can process entered data.

Create a manual application plan

You use the Manual application plan to map data to documents or records without using an agent, Interact client, or business application. To create a manual application plan, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans** and select **Manual**.
3. In the right pane, on the **Applications** tab, select **New**.
4. In the **Plan Settings** dialog box, select the **General** tab.
5. Under **Information**, complete the following substeps.
    1. Next to **Content Type**, select either **Document** or **Record**.

        > **Note:** You cannot change the content type once the application plan is created.

    2. In the **Name** box, type a unique name for your application plan.
    3. Optional. In the **Description** box, type a description for your application plan.
    4. Clear the **Is active** check box if you do not want this application plan to be active.
6. Click **OK**.

Mobile

About the Mobile type

A mobile application plan allows you to associate property values to documents that contain images that you can upload from a mobile device.

A user accesses a mobile application plan from an Perceptive Content interface on a mobile device. The mobile plan allows a user to create documents or folders that are indexed with properties on the server. A user can upload images, which become pages, associated with those documents or folders.

Create a mobile application plan

To create an application plan for a mobile device, complete the following steps.

To use mobile application plans, an Interact Mobile license is required.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. In the left pane, expand **Application Plans** and select **Mobile**.
3. In the right pane, on the **Application Plans** tab, select **New**.
4. In the **Plan Settings** dialog box, select the **General** tab.
5. Under **Information**, complete the following substeps.
    1. In the **Name** field, type a unique name for your application plan.
    2. Optional. In the **Description** field, enter text to describe this application plan.
    3. Optional. Clear the **Is active** check box if you do not want this application plan to be active.

Work with application maps

What is an application plan map?

An application plan map specifies where to store an item in Perceptive Content and which property values to assign to the item. An item can be in the form of a document, record, shortcut, or folder.

An application plan map assigns the location and property values to an item during capture and other processing tasks, such as when copying or redacting a document.

When you define a property for an item, you must specify which value to assign to the property. You must also specify the source from which to gather the value for each property.

When you define a location that includes folders, Perceptive Content creates any folders that do not already exist to construct the specified path.

About using legacy document and project maps

In earlier versions of Perceptive Content, you could use two map types, document and project, to define documents and projects using an application plan. In this version, you use a single map to define documents, folders, and shortcuts.

When you upgrade from an earlier version of Perceptive Content, the system automatically converts your existing document and project maps according to the following guidelines:

- `Document maps`. When you have an existing document map, Perceptive Content converts it to a document map where the drawer and document properties remain the same. After you convert a map, you can begin using folder hierarchy by assigning a document name or adding folder levels to that document map.
- `Project maps`. If a project map exists on your current system, Perceptive Content converts it to a shortcut map with two levels of folders and assigns a drawer. In the first level, the Folder Name value is set to the project type used in the source project map and the Folder Type value is set to "Legacy folder," a system folder that Perceptive Content uses to migrate legacy project types. You cannot modify the first level legacy folder level. In the second folder level of the shortcut map, Perceptive Content assigns the name, type, and any properties set in the source project map. You can modify the second folder level.

In addition to creating levels, Perceptive Content assigns the "Folders" drawer to the shortcut map.

Create an application plan document map

To create an application plan document map, complete the following steps.

1. Define an application plan map drawer.
2. Define an application map folder level. You may define more than one folder level.
3. Define an application plan map document level.
4. Optional. Define an application map shortcut.
5. Define an application map drawer.
6. Test an application plan map.

Modify an application plan map for a document

You can modify an application map for documents by modifying drawer attributes, by adding, moving, or removing folder levels, and by modifying or resetting map attribute values. To modify an application plan map, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application plan you want to modify.
4. Click **Modify**.
5. Perform any of the following actions.

Situation	Steps
Add a folder level	1. Click **Add folder** level. 2. To define the folder, in the folder level box, double-click the **Folder Name** and **Folder Type** rows and set the values you want.
Move a folder level	• To move a folder level, in that level, click the **Move Up** or **Move Down** buttons.
Remove a folder level	• To delete a level from the map, click the **Remove Level** button.
Add a document level	1. Click **Add document level**. 2. To define the document, in the **Document level** box, double-click the property rows and set the values you want.
Modify a value	• To edit a value, select the value row and click the **Modify Row** button.
Reset a value	1. To reset all of the values in a folder or document properties level, in the level header, click the **Reset Level** button. 2. To reset a value in a row, select the value row and click the **Reset Row** button.

Note: If you remove the document level, Perceptive Content converts the document map to a folder map.

6. Save the application plan.

Modify a document property in a map

In an application plan, you define a type for each document property that specifies the type of information that you want Perceptive Content to store. To modify the settings in a LearnMode application plan map that defines how to assign a property to a document during the capture process, complete the following steps.

1. In **Application Plan Designer**, click the **Map** tab.
2. In the **Screens** pane, select the screen for which you want to modify a document property.
3. Select the property you want to modify and click **Modify**.
4. In the **<Property> Attributes** dialog box, modify the property as you normally would by updating the

general and advanced attributes.

5. Click **OK** to close the **<Property> Attributes** dialog box.

Modify an application plan map for records

You can modify an application map for records at each map level. The map levels are file plan, record category, record folder, and record. You can add, move, and remove only the record folder level. You can modify and reset attribute values for any level. To modify an application plan map, complete the following steps.

1. In **Management Console**, in the left pane, under **Select Department**, select a department from the list.
2. Expand **Application Plans** and select an application plan type.
3. In the right pane, on the **Applications** tab, select the application plan you want to modify.
4. Click **Modify**.
5. To modify a level of an application plan map, choose from the following steps.
   - To add a record level, click the **Add record folder level** button.
   - To move a record folder level up or down, click the **Move Up** button or **Move Down** button.
   - To remove a level, click the **Remove Level** button.
   - To modify an attribute value, select the attribute row, then click the **Modify Row** button.
   - To reset an attribute value, select the attribute row, then click the **Reset Row** button.
   - To reset all attribute values in a record category level, record folder level, or record level, click the **Reset Level** button.

Next Test your application plan map.

Test an application plan map

To test an application plan map, complete the following steps.

1. Choose the application plan type to which your map applies.

Application Plan Type	Steps
Agent, **Interact**, **External**, **Manual**	1. In **Application Plan Designer**, on the **Map** tab, click the **Test Map** arrow. 2. In the **Map Test** dialog box, verify that the system returns the correct result for each property.
LearnMode	1. Start your business application and open the window or screen that you want to test. 2. Navigate to an active business object.

Application Plan Type	Steps
	3. In **LearnMode Application Plan Designer**, in the **Screens** pane, select the screen for which you want to test a map. 4. Click the **Map** tab and click the **Test Map**  arrow. 5. In the **Map Test** dialog box, verify that the system returns the correct result for each property. 6. To test another business object, navigate to the new business object in your application and click **Refresh**.

When testing **Agent**, **Interact**, or **External** application plan maps, the system returns the example data you set in each dictionary data element. If example data does not exist, the system returns null values.

2. Close the **Test Map** dialog box.

Map Components

Map attribute values for a document

When you create a map for an application plan, you can define folder names, folder types, folder custom properties, document properties, document types, and document custom property attributes for that plan. The attributes you can define depend on the application plan type.

Document custom property types

This table lists the document custom property attributes you can define for an application plan map.

Document Custom Property Source	Define the Document Custom Property Value
Barcode	By the name you defined in an external application.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the custom property.
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.

Document Custom Property Source	Define the Document Custom Property Value
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Document types

This table lists the document type attributes you can define for an application plan map.

Document Type Source	Define the Document Type Value
Barcode	By the name you defined in an external application.
Document Type	By the selected document type.
Document Type List	By the user's selection in a predefined document type list.
Dictionary	By the selected data element defined in the Dictionary.

Document fields

This table lists the field attributes you can define for an application plan map.

Field Source	Define the Field Value
Barcode	By the name you defined in an external application.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the field.

Field Source	Define the Field Value
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Folder names

This table lists the folder name attributes you can define for an application plan map.

Folder Name Source	Define the Folder Name Value
Barcode	By the name you defined in an external application.
Document Key	By the selected document key.
Document Custom Property	By the selected custom property if the document type associated with this map has a custom property.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the folder.
Literal	By the value you enter.
User Entry	By the value the user enters.

Folder Name Source	Define the Folder Name Value
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Folder types

This table lists the folder type attributes you can define for an application plan map.

Folder Type Source	Define the Folder Type Value
Barcode	By the name you defined in an external application.
Document Key	By the selected document key.
Document Custom Property	By the selected custom property if the document type associated with this map has a custom property.
Folder Type	By the selected folder type.
Folder Type List	By the user's selection in a predefined folder type list.
Dictionary	By the selected data element defined in the Dictionary.

Folder custom property types

This table lists the folder custom property attributes you can define for an application plan map.

Folder Custom Property Source	Define the Folder Custom Property Value
Barcode	By the name you defined in an external application.
Document Key	By the selected document key.
Document Custom Property	By the selected custom property if the document type associated with this map has a custom property.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the custom property.
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Map attribute values for a record

When you create a map for a record application plan, you define attributes for each of its levels.

File Plan

This table lists the file plan attributes you can define for an application plan map.

File Plan Source	Define the File Plan Value
System File Plan	By the selected file plan.

Record Category

This table lists the record category attributes you can define for an application plan map.

Record Category Source	Define the Record Category Value
Barcode	By the name you defined in an external application.
System Record Category	By the selected record category.
Dictionary	By the selected data element defined in the Dictionary.

Record Folder Name

This table lists the record folder name attributes you can define for an application plan map.

Record Folder Name Source	Define the Record Folder Name Value
Barcode	By the name you defined in an external application.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the folder.
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Record Folder Type

This table lists the record folder type attributes you can define for an application plan map.

Record Folder Type Source	Define the Record Folder Type Value
Barcode	By the name you defined in an external application.
Record Folder Type	By the selected folder type.
Dictionary	By the selected data element defined in the Dictionary.

Record Folder Custom Properties

This table lists the record folder custom properties you can define for an application plan map.

Record Folder Custom Property Source	Define the Record Folder Custom Property Value
Barcode	By the name you defined in an external application.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the custom property.
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Record Properties

This table lists the record property attributes you can define for an application plan map.

Record Property Source	Define the Record Property Value
Barcode	By the name you defined in an external application.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the property.
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Record Type

This table lists the record type attributes you can define for an application plan map.

Record Type Source	Define the Record Type Value
Barcode	By the name you defined in an external application.
Record Type	By the selected record type.
Dictionary	By the selected data element defined in the Dictionary.

Record Custom Properties

This table lists the record custom property attributes you can define for an application plan map.

Record Custom Property Source	Define the Record Custom Property Value
Barcode	By the name you defined in an external application.
Dictionary	By the selected data element defined in the Dictionary.
Undefined	Perceptive Content does not assign a value to the custom property.
Literal	By the value you enter.
User Entry	By the value the user enters.
Predefined List	By the user's selection in a predefined list.
Username (Current)	By the current user name.
Username (Select)	By a selection in a set of user names.
Current Date and Time	By the current date and time.
Sequence Number	By a number in a sequence.
Unique ID	By a unique ID that is automatically assigned.

Define an application plan map drawer

In an application plan, you define a single drawer for a document or folder map. To define the drawer your application plan map uses to store documents, folders, and shortcuts, complete the following steps.

1. Optional. In **Application Plan Designer**, if using the **LearnMode** application type, in the **Screens** pane, select the screen to which you want to add a map.
2. In **Application Plan Designer**, in the right pane, click the **Map** tab and perform one of the following actions.

Situation	Steps
Define a document map	• Under **Document Map**, in the **Drawer** level, double-click the drawer row.

Situation	Steps
Define a folder map	• Under **Folder Map**, in the **Drawer** level, double-click the drawer row.
Define a shortcut map	• Click **Add shortcut map**.

3. In the**Drawer Attributes** dialog box, under **General**, perform one the following actions.
   - To use a system drawer, in the **Source** list, click **System Drawer** and in the **Drawer** list, select a drawer.
   - To prompt a user to select a drawer during the linking process, in the **Source** list, click **System Drawer** List.
   - To set the drawer using a dictionary element defined for this screen, in the **Source** list, click **Dictionary** and in **Drawer**, select an element.
4. Optional. Under **Advanced**, perform one or more of the following actions.
   - Create a Visual Basic script.
   - To allow a user to modify the drawer value, select **User modifiable**, if available.
   - If you selected **User modifiable** and you want Perceptive Content to remember the latest user modified value, select **Remember last value**.
   - To set the drawer value to overwrite the proposed key, select **Overwrite proposed key**.
5. To close the **Drawer Attributes** dialog box, click **OK**.

Define an application plan map file plan

You can store a record in Perceptive Content by using an application plan map to define a file plan level. The file plan level is the highest level of organization in an application plan map. You must define a file plan level before you define a record category level. To define a file plan, complete the following steps.

1. In **Application Plan Designer**, select the **Map** tab.
2. In the **File Plan** level, under **Property**, double-click **File Plan**.
3. In the **File Plan Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the file plan value.
4. In the **File Plan** list, select a file plan.
5. Optional. In the **File Plan Attributes** dialog box, under **Advanced**, perform one or more of the following actions.
   - To allow the user to modify the file plan value, select **User modifiable**, if available.
   - If you selected **User Modifiable** and you want Perceptive Content to remember the last modification to the field name, select **Remember last value**.
   - To set the file plan value to overwrite a value proposed in the capture process, select **Overwrite proposed value**.
6. To close the **File Plan Attributes** dialog box, click **OK**.

Define an application plan map document level

The document level is always the last level in a document map. You must define a document type and at least one other property (such as Field1 or a custom property) for a document map. To assign the metadata properties for the document you create when linking with this application plan, complete the following steps.

1. In **Application Plan Designer** select the **Map** tab.
2. In the **Document** level, under **Property**, complete the following steps:

Situation	Steps
Define the document name	1. Double-click the **Document name** row. 2. In the **Document Name Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the document name value. 3. In the **Value** list, select a document name value.
Define the document keys	1. Double-click the document key row you want to define. 2. In the **<Field name> Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the document key value. 3. In the **Value** list, select a document key value.
Define the document type	1. Double-click the **Document type** row. 2. In the **Document Type Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the document type value. 3. In the **Value** list, select a document type value.
Define a document custom property	1. Double-click the custom property row. 2. In the **<Field name> Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the document key value. 3. In the **Value** list, select a document type value.

3. Optional. Under **Advanced**, select any advanced option available for the property source type.

4. Click **OK**.
5. Repeat these steps to assign additional properties.

Define an application map folder level

Folder levels allow you to store folders using the folder hierarchy that exists on your system. To map one or more folders to create a path in which to store documents, folders, and shortcuts you create with this application plan, complete the following steps.

1. In **Application Plan Designer** select the **Map** tab.
2. Click **Add folder level**.
3. Under **Property**, complete the following steps.

Situation	Steps
Define the folder name	1. Double-click **Folder name**. 2. In the **Folder Name Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the folder name value. 3. In the **Value** list, select a folder name value.
Define the folder type	1. Double-click **Folder Type**. 2. In the **Folder Type Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the folder type value. 3. In the **Value** list, select a folder type value.
Define a folder custom property	1. In the **Folder** level, under **Property**, double-click the custom property you want to define. 2. In the **<Custom Property Type> Attributes** dialog box, select a folder custom property.

4. Optional. Under **Advanced**, select any advanced option available for the property source type.
5. Click **OK**.
6. Repeat these steps to create additional folder levels.

Define an application plan map record category level

You can store a record in Perceptive Content by using an application plan map to define a record category level in a file plan level. To define a record category level, complete the following steps.

Prerequisite You must define a file plan level before defining a record category level.

1. In **Application Plan Designer**, select the **Map** tab.
2. In the **Record Category** level, under **Property**, double-click **Record Category**.
3. In the **Record Category Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the record category value.
4. In the **Record Category** list, select a record category.

   Note: If you have not already defined a file plan level, the record category list will not contain any values.

5. Optional. In the **Record Category Attributes** dialog box, under **Advanced**, perform one or more of the following actions.
   - Create a Visual Basic script for the record category value.
   - To allow the user to modify the record category value, select **User modifiable**, if available.
   - If you selected **User Modifiable** and you want Perceptive Content to remember the last modification to the field name, select **Remember last value**.
   - To set the record category value to overwrite a value proposed in the capture process, select **Overwrite proposed value**.

Define an application plan map record folder level

You can store a record in Perceptive Content by using an application plan map to define a record folder level in a record category level or in other record folder levels. You define the record folder name, record folder location, and record folder type. You can also define record folder custom properties if you have previously created them and assigned them to the record folder type. To define a record folder level, complete the following steps.

1. In **Application Plan Designer**, select the **Map** tab.
2. To create a record folder level, click **Add record folder level**.
3. Define the record folder name.
   1. In the **Record Folder** level, under **Property**, double-click **Record Folder Name**.
   2. In the **Record Folder Name Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the record folder name value.
   3. In the **Value** list, select a record folder name value.
4. Define the record folder type.
   1. In the **Record Folder** level, under **Property**, double-click **Record Folder Type**.
   2. In the **Record Folder Type Attribute** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the record folder type value.
   3. In the **Value** list, select a record folder type value.
5. Define a record folder custom property.
   1. In the **Record Folder** level, under **Property**, double-click the custom property you want to define.
   2. In the **<Custom Property Type> Attributes** dialog box, select a record folder custom property.
6. Optional. In any **Attributes** dialog box, under **Advanced**, perform one or more of the following actions.
   - Create a Visual Basic script for a document value.

- To allow the custom property to remain empty, for all source options except **Undefined**, select **Allow empty**.
- To allow a user to modify the name field, for all source options except **Undefined**, **User Entry**, **Predefined List**, **Username (Select)**, and **Sequence Number**, select **User modifiable**.
- To have Perceptive Content remember the last modification to the field name, for all source options except **Undefined**, select **Remember last value**.
- To have the mapped field name overwrite a value proposed in the capture process, for all source options except **Undefined**, select **Overwrite proposed key**.

7. To close any **Attributes** dialog box, click **OK**.
8. Repeat these steps to create additional folder levels.

Define an application plan map record level

You can store a record in Perceptive Content by using an application plan map to define a record level in a record folder level. You define the record name, record type, and mandatory record properties. You can also define record custom properties if you have previously created them and assigned them to the record type. To define a record level, complete the following steps.

1. In **Application Plan Designer**, select the **Map** tab.
2. Define the record name.
    1. In the **Record** level, under **Property**, double-click **Name**.
    2. In the **Name Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the name value.
    3. In the **Value** list, select a name value.
3. Define the record properties. The following record properties are mandatory: **Author**, **Originating Organization**, and **Publication Date**. The following properties are also mandatory if the record is correspondence: **Addressees**, and **Other Addressees**.
    1. In the **Record** level, under **Property**, double-click the property you want to define.
    2. In the **<Field name> Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the property value.
    3. In the **Value** list, select a property value.
4. Define the record type.
    1. In the **Record** level, under **Property**, double-click **Record Type**.
    2. In the **Record Type Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the record type value.
    3. In the **Value** list, select a record type value.
5. Define the record custom properties.
    1. In the **Record** level, under **Property**, double-click the record custom property you want to define.
    2. In the **<Field Name> Attributes** dialog box, under **General**, from the **Source** list, select the source you want to use to assign the property value.
    3. In the **Value** list, select a property value.
6. Optional. In any **Attributes** dialog box, under **Advanced**, perform one or more of the following actions.

- Create a Visual Basic script for a record value.
- To allow the field to remain empty, for all source options except **Undefined**, select **Allow empty**.
- To allow a user to modify the field name, for all source options except **Undefined**, **User Entry**, **Predefined List**, **Username/ (Select)**, and **Sequence Number**, select **User modifiable**.
- To have Perceptive Content remember the last modification to the field name, for all source options except **Undefined**, select **Remember last value**.
- To have the mapped field name overwrite a value proposed in the capture process, for all source options except **Undefined**, select **Overwrite proposed key**.

7. To close any **Attributes** dialog box, click **OK**.

Define an application map shortcut

When you capture and store your document, you save it to a specific folder. You can create a shortcut to that folder. A shortcut is a pointer to the document from other folders in your Perceptive Content environment. A shortcut allows you to reference the document in multiple folders but maintain the integrity of the document in its original location. To define a shortcut in an application plan map, complete the following steps.

1. In **Application Plan Designer**, on the **Map** tab, under **Shortcut Map**, click **Add shortcut map**.
2. Define a drawer and folder level for your shortcut destination.
3. Optional. Add additional folder levels.

Result Your shortcut appears as a document in an ImageNowExplorer folders view.

Test an application plan map

To test an application plan map, complete the following steps.

1. Choose the application plan type to which your map applies.

Application Plan Type	Steps
Agent, **Interact**, **External**, **Manual**	1. In **Application Plan Designer**, on the **Map** tab, click the **Test Map** arrow. 2. In the **Map Test** dialog box, verify that the system returns the correct result for each property.
LearnMode	1. Start your business application and open the window or screen that you want to test. 2. Navigate to an active business object. 3. In **LearnMode Application Plan Designer**, in the **Screens** pane, select the screen for which you want to test a map. 4. Click the **Map** tab and click the **Test Map** arrow.

Application Plan Type	Steps
	5. In the **Map Test** dialog box, verify that the system returns the correct result for each property. 6. To test another business object, navigate to the new business object in your application and click **Refresh**.

When testing **Agent**, **Interact**, or **External** application plan maps, the system returns the example data you set in each dictionary data element. If example data does not exist, the system returns null values.

2. Close the **Test Map** dialog box.

Troubleshoot application plan testing

If you encounter error messages while testing your application plan or application plan map, try any of the following possible resolutions.

I receive an error when I test my application plan map

Error	Resolution
Required value	Verify whether required values have been left blank. To enable the Map test, enter a value where required, or modify your application plan as needed.
The folder type does not match the type set for the folder structure. Select a different folder type.	Verify whether the folder type is valid by accessing the folder type hierarchy in Management Console.
The document type does not match the type set for the folder structure. Select a different document type.	Verify whether the document type is valid in the hierarchy specified for the drawer in Management Console.

ImageNow cannot find my application plan screen

Cause	Resolution
The appropriate screen is not being displayed within the business application.	Switch to the appropriate screen within the business application and test again.
The screen identifier used to create the application plan screen is not present.	Check the screen identifier within LearnMode. You might need to rebuild the application plan screen using a different screen identifier.

Cause	Resolution
The screen identifier being used is not static and/or not unique to the current business application screen.	Check the screen identifier within LearnMode. You might need to rebuild the application plan screen using a different screen identifier.

ImageNow cannot locate my business application

Cause	Resolution
The business application is not running.	Start the business application and go to the appropriate screen.
The name of the application as it appears in the title bar of the business application window has changed since creating the application plan.	Open the LearnModeapplication plan and verify that the window name matches the name of the business application window. You might need to modify this value.
The name of the business application window may change for each user or for each login session.	Some applications allow you to modify the title of the window. Check to see if such settings exist within the business application or if you can use a wildcard.

ImageNow cannot retrieve data from my clipboard

Cause	Resolution
The appropriate connect strings are not being used.	Open the LearnModeapplication plan and verify that the correct connect strings are being used. Be sure that each connect string segment ends with a semicolon (;). Verify that the connect strings you are using are valid.
The operation that attempts to copy the data from the business application is happening too fast.	Make changes to the Retry/Delay settings to change the timing of the data capture.
The terminal emulation software being used to connect to the business application is not the same as the emulator used when creating the application plan.	Make sure that the same terminal emulation software is being used on each computer running ImageNow Client. Be sure that all the versions are the same; different versions will sometimes behave differently.

The capture profile and application settings conflict

Cause	Resolution
The settings in the capture profile and the selected application plan specify to store the captured files in different locations.	Select an application plan that does not conflict with the capture profile.
The capture profile specifies to store the captured files in a folder, but the selected application plan specifies to store the files in a different folder or in a drawer, or the application plan stores the document in a folder, but the capture profile does not have the same setting.	Modify the settings in the capture profile or the application plan so that the defined location does not conflict. For example, if you define the application map to store the captured files in a folder, ensure that you save documents to a folder in the capture profile.

I cannot process any additional HyperLearn fonts

Cause	Resolution
The maximum number of fonts (50) have been defined.	To process more fonts, you need to remove at least one of the existing fonts. Doing so disables any application plans using that font. Keeping track of the fonts your application plans use is the best way to avoid this. If you get a message that the font has been processed, the font is stored in Perceptive Content.

Work with the dictionary

What is the dictionary?

The dictionary lists all the data elements available within a single application plan.

When adding screens to your LearnMode type application plan, data elements are the screen elements captured from the business application. You can add and manage the data elements for the current application plan. You also can rearrange, rename, delete, and test the data elements.

You can reuse a data element for more than one business application screen. For example, if you have an account number on your first business application screen and second business application screen, you only need to define it once.

Add a data element to the dictionary

A data element is a piece of information collected from your associated business application screen. To add a data element to the dictionary, complete the following steps.

1. In the **Dictionary** pane, click the **New Field** button.
2. In the **Dictionary Field** dialog box, under **General**, complete the following actions.
    1. In the **Name** box, type a unique name for your new data element.
    2. Under **Processing**, add a macro or a script.
3. Optional. In the **Dictionary Field** dialog box, under **Sample**, complete one of the following actions.
    - If you selected a macro as your processing type, type an example of the unaltered data in the **Input** box.
    - If you selected a script as your processing type, type an example of the unaltered data in the **Input** box. Click **Refresh** to update the results and verify the script returns the correct data.
4. Click **OK** to save your changes.

Define a dictionary data element using Internet Explorer

A data element is a piece of information collected from your associated business application screen. To define a dictionary data element using Internet Explorer, complete the following steps.

Prerequisite Before you add a data element to the dictionary, you need to create a screen. You also need to ensure you have your web-based business application open to the screen and business object you want to capture.

1. In **Application Plan Designer**, on the **Elements** tab, select Internet Explorer from the Method list.

    The system captures the controls for the business application screen and displays a tree representation in **Application Plan Designer**.
2. Scroll down the list of controls from the business application screen.

    Note: Each row of the displayed tree represents an available control. The left column lists the type of control. The right column displays the current data for the control. The screen identifier is the data on your business application screen that uniquely identifies the screen.
3. Do one of the following actions to add a data element:
    - If the data element is already added to the dictionary, point to the appropriate control you want to define, and then drag it to the appropriate data element in the **Dictionary** pane.
    - If the data element is not in the dictionary, point to the control you want to add and define, and then drag it to the bottom of the data element list in the **Dictionary** pane.
4. In the **Dictionary Field** dialog box, add the data element as you normally would.

Define a dictionary data element using Terminal Capture

A data element is a piece of information collected from your associated business application screen. To define a dictionary data element using a Terminal Capture method, complete the following steps.

Prerequisite Before you add a data element to the dictionary, you need to create a screen. You also need to ensure you have your business application open to the screen and business object you want to capture.

1. In **Application Plan Designer**, on the **Elements** tab, select **Terminal Capture** from the **Method** list.
2. Click **Settings**.
3. In the **Terminal Capture Settings** dialog box, complete the following substeps.
    1. In the **Connect Strings** box, type the connect string.
    2. In the **Retries** box, type the number of times you want the system to attempt to connect to the terminal emulation program.
    3. In the **Delay** (milliseconds) box, type the length of time you want the system to wait between connection attempts.
4. The system copies the screen contents of the screen and displays it in **LearnMode**.
5. To add a data element, perform one of the following actions.
    - If the data element is already added to the dictionary, use the mouse to drag a box around the data in the captured window that you want to define, and then drag it to the appropriate data element in the **Dictionary** queue.
    - If the data element is not in the dictionary, use the mouse to drag a box around the data in the captured window that you want to define, and then drag it to the bottom of the data element list in the **Dictionary** queue.
6. In the **Dictionary Field** dialog box, add the data element as you normally would.
7. Repeat these steps for each data element you want to define.

Define a dictionary data element using Viewpoint

A data element is a piece of information collected from your associated business application screen. To define a dictionary data element using a Viewpoint method, complete the following steps.

Prerequisite Before you add a data element to the dictionary, you need to create a screen. You also need to ensure you have your business application open to the screen and business object you want to capture.

1. In **Application Plan Designer**, on the **Elements** tab, select **Viewpoint** from the **Method** list.
2. Click **Capture**.

    The system captures a screen shot of the business application window and displays it in **Application Plan Designer**.
3. Perform one of the following actions to add a data element.
    - If the data element is already added to the dictionary, point to the business application field you want to define, and then drag it to the appropriate data element in the **Dictionary** pane.
    - If the data element is not in the dictionary, point to the business application field you want to add and define, and then drag it to the bottom of the data element list in the **Dictionary** pane.

4. In the **Dictionary Field** dialog box, add the data element as you normally would.
5. Repeat these steps for each data element you want to define.

Define a dictionary data element using HyperLearn

A data element is a piece of information collected from your associated business application screen. To define a dictionary data element using a HyperLearn method, complete the following steps.

Prerequisite Before you add a data element to the dictionary, you need to create a screen. You also need to ensure you have your business application open to the screen and business object you want to capture.

When defining a data element, you can use the LEFT ARROW key, the RIGHT ARROW key, the DOWN ARROW key, or the UP ARROW key to move the box. To resize the box, press SHIFT+LEFT ARROW. You can also use the mouse to move the box or resize it.

1. In **Application Plan Designer**, on the **Elements** tab, select **HyperLearn** from the **Method** list.

   The system captures a screen shot of the business application window and displays it in **Application Plan Designer**.
2. Click **Settings** to add or modify the fonts used by the business application.

   Note: If you do not select the correct font, you will get an error message when trying to define a dictionary data element, or document property value. If this occurs, you must repeat this step until you are able to successfully define a property.
3. Click the mouse button and drag a box around the data element containing the value in the captured window. Extend the box to the end of the control to ensure that the longest value entered for that control will be entered into the dictionary data element and do one of the following actions:
   - If the data element is already added to the dictionary, drag it to the appropriate data element in the **Dictionary** pane.
   - If the data element is not in the dictionary, drag it to the bottom of the data element list in the **Dictionary** pane and then modify it as you normally would.

Define a dictionary data element using Window Walker

A data element is a piece of information collected from your associated business application screen. To define a dictionary data element using a Window Walker method, complete the following steps.

Prerequisite Before you add a data element to the dictionary, you need to create a screen. You also need to ensure you have your business application open to the screen and business object you want to capture.

1. In **Application Plan Designer**, on the **Screen Elements** tab, select **Window Walker** from the **Method** list.

   Note: A list of window controls displays in the captured window. Expand the list as needed to show the various controls from the program.

   The system captures a tree that contains the controls on the business application window and displays it in **Application Plan Designer**.

2. Click the **PLUS SIGN (+)** to expand the tree.

   Note: Each row in the tree represents an available control. The first part of the row lists the type of control. The second part of the line (to the right of the arrow) displays the current data for the control.

3. To add a data element, complete one of the following actions.
   - If the data element is already added to the dictionary, point to the business application control you want to define, and then drag it to the appropriate data element in the **Dictionary** pane.
   - If the data element is not in the dictionary, point to the business application control you want to add and define, and then drag it to the bottom of the data element list in the **Dictionary** pane.
4. In the **Dictionary Field** dialog box, add the data element as you normally would.
5. Repeat these steps for each data element you want to define.

Modify a data element in the dictionary

A data element is a piece of information collected from your associated business application screen. To modify a data element in the dictionary, complete the following steps.

1. In **Application Plan Designer**, in the **Dictionary** pane, select the data element you want to modify.
2. Click the **Modify Field** button.
3. In the **Dictionary Field** dialog box, modify the data element as you normally would, by changing its name, applying a different processing type, or modifying the data sample.
4. Click **OK** to save your changes.

Remove a data element from the dictionary

To remove a data element from the dictionary, complete the following steps.

1. In **Application Plan Designer**, in the **Dictionary** pane, select the data element you no longer want to define.
2. Click the **Remove from Current Screen** button.
3. In the confirmation dialog box, click **Yes**.

Delete a data element in the dictionary

To delete a data element in the dictionary for all defined screens, complete the following steps.

1. In **Application Plan Designer**, in the **Dictionary** pane, select the data element you want to delete.
2. Click the **Delete Field** button.
3. In the confirmation dialog box, click **Yes**.

Test data dictionary elements

A data element is a piece of information collected from your associated business application screen. To test defined data elements in the dictionary, select the application plan type and complete the following steps.

1. If you use a GUI-based application plan, complete the following steps.
    1. Start your business application.
    2. Open the window or screen that you want to test.
    3. Navigate to an active business object.
2. In **Application Plan Designer**, in the **Dictionary** pane, click the **Test Dictionary** button.
3. In the **Test Results <Screen name>** dialog box, verify the system returns the correct results for each dictionary data element.

    When testing agent application plans, the system returns the example data you set in each dictionary data element. If example data has not been added, the system will return null values.
4. If you want to test another business object in a GUI-based application, navigate to the new record and click **Refresh**.
5. Close the **Test Results <Screen name>** dialog box.

Work with predefined lists

What is a predefined list?

A predefined list is a group of values for a document or a record property. A user can select a predefined list from a list box.

Similar to a user entry document or record property, a predefined list requires some user interaction. In this case, the user selects an item from a predefined list of options. You can set up and maintain multiple predefined lists; however, only one list can be used with each defined property.

Create a predefined list

Predefined lists are a group of values that a user can select for item properties. To create a predefined list in an application plan, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the screen for which you want to modify map values.
2. In the right pane, click the **Map** tab, and perform one of the following actions.

Situation	Steps
Document Level	• Double-click **Document Name**, **Field1**, **Field2**, **Field3**, **Field4**, or **Field5**.

Situation	Steps
Folder Level	• Double-click **Folder Name**.
Record Level	• Double-click **Name**, **Field1**, **Field2**, **Field3**, **Field4**, **Field5**, **Author**, **Originating Organization**, **Addressees**, **Other Addressees**, **Media Type**, or **Format**.
Record Folder Level	• Double-click **Record Folder Name** or **Record Folder Location**.

3. In the **<Property> Attributes** dialog box, in the **Source** list, select **Predefined List**.
4. In the **Value** list, select **Edit Lists**.
5. In the **Predefined Lists** dialog box, click **New**, and type a name for the list.
6. To add the list description and list member values, complete the following substeps.
   1. Select the list name and click **Modify**.
   2. In the **Edit Predefined List** dialog box, on the **General** tab, in the **Description** box, type a description of the list.
   3. On the **List Members** tab, in the **Name** box, type the first value to appear in the list.
   4. Click **Add**.
   5. Repeat the previous substeps until you add all list member values.

   Note: Click the **Name** header to sort list member values in ascending or descending order.
7. Click **OK** and **Close**.
8. In the **<Property> Attributes** dialog box, in the **Value** list, select the new predefined list.
9. Click **OK** to return to the **Application Plan Designer**.

Modify or rename a predefined list

Predefined lists are groups of values that a user can select for item properties. To modify or rename a predefined list in an application plan, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the screen for which you want to modify map values.
2. In the right pane, click the **Map** tab, and perform one of the following actions.

Situation	Steps
Document Level	• Double-click **Document Name**, **Field1**,

Situation	Steps
	Field2, **Field3**, **Field4**, or **Field5**.
Folder Level	• Double-click **Folder Name**.
Record Level	• Double-click **Name**, **Field1**, **Field2**, **Field3**, **Field4**, **Field5**, **Author**, **Originating Organization**, **Addressees**, **Other Addressees**, **Media Type**, or **Format**.
Record Folder Level	• Double-click **Record Folder Name** or **Record Folder Location**.

3. In the **<Property> Attributes** dialog box, in the **Source** list, select **Predefined List**.
4. In the **Value** list, select **Edit Lists**.
5. In the **Predefined Lists** dialog box, select the list you want to modify, and click **Modify**.
6. Optional. On the **General** tab, modify the name or the description of the predefined list.
7. Optional. To add or modify list member values, on the **List Members** tab, perform any of the following actions.
   - To add a list member, in the **Name** box, type a list member value, and then click **Add**.
   - To change the location of a list member value, select the value, and then click **Move Up** or **Move Down**.
   - To remove a value from the list, select the value, and then click **Remove**.
   - To sort list member values in ascending or descending order, click the **Name** header.
8. Click **OK**, **Close**, and **OK** to return to the **Application Plan Designer**.

Delete a predefined list

To delete a predefined list, complete the following steps.

1. In **Application Plan Designer**, click the **Map** tab.
2. In the **Screens** pane, select the screen for which you want to delete the predefined list and complete one of the following actions.

Situation	Steps
Document Level	• Double-click **Document Name**, **Field1**, **Field2**, **Field3**, **Field4**, or **Field5**.
Folder Level	• Double-click **Folder Name**.

Situation	Steps
Record Level	• Double-click **Name**, **Field1**, **Field2**, **Field3**, **Field4**, **Field5**, **Author**, **Originating Organization**, **Location**, **Addressees**, **Other Addressees**, **Media Type**, or **Format**.
Record Folder Level	• Double-click **Record Folder Name** or **Record Folder Location**.

3. In the **<Property> Attributes** dialog box, in the **Source** list, select **Predefined List**.
4. In the **Value** list, select **Edit Lists**.
5. In the **Predefined Lists** dialog box, select the name of the list you want to delete.
6. Click **Delete**.
7. In the confirmation dialog box, click **Yes**.
8. Click **Close** and **OK** to return to the **Application Plan Designer**.

Work with screens

What is a screen?

A screen represents a screen in your business application that is learned by applying one of the LearnMode methods, such as HyperLearn or ViewPoint.

When you create a screen, you capture data from the business application screen to create document keys. You can also add screens to an existing application plan, make changes to the way an existing screen links to the business application, delete a screen, and rename a screen.

In Perceptive Content, you can use Microsoft Visual Basic Scripting to identify specific screens of your business application when multiple screens are open. You can create, modify, and apply these scripts when adding or editing a screen.

Add a screen

A screen is an image captured from your business application that you add to your LearnMode type application plan to collect data and create document keys or record keys. To add a screen to a LearnMode application plan, complete the following steps.

Prerequisite Before you add a screen, you must start your business application and navigate to the screen you want and an active business object.

1. In **Application Plan Designer**, in the **Screens** pane, click the **Add New Screen** button.
2. Under **Name**, double-click the new screen and type a unique name.
3. In the **Screens** pane, click the **Move Up**

 or **Move Down**  button to change the order of the screens. For example, you can move the first screen in your business application to the top of the list.

4. Repeat the steps above for each screen you want to add to your application plan.

Duplicate a screen

A screen is an image captured from your business application you add to your LearnModetype application plan to collect data and create item property values. To duplicate a screen, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the screen you want to duplicate.
2. Click the **Duplicate Screen**  button.

Delete a screen

To delete a screen in an application plan, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the screen you want to delete.
2. Click the **Delete Screen**  button.
3. In the confirmation dialog box, click **Yes**.

Rename a screen

A screen is an image captured from your business application you add to your LearnModetype application plan to collect data and create property values. To rename a screen, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, double-click the screen you want to rename.
2. In the **Name** field, type the new screen name and click **OK**.

Add a script to a screen

A script defines how Perceptive Content formats and processes screen data elements before the elements are added to the dictionary. To add a script to a screen, complete the following steps.

Prerequisite Before you add a script to a screen, you must create a screen. Start your business application and navigate to the screen you want.

The ScreenFound variable value is False by default. The script sets the value to True if the screen is found. If False, the next available screen will be used.

1. In **Application Plan Designer**, in the **Screens** pane, click the **Add New Screen** button.
2. In the **Screen Properties** dialog box, in the **Name** field, type a unique name for the screen.
3. Under **Identification Method,** select **Script**.
4. In the **Script** list, select a script. If no scripts appear, select **Manage Scripts** and perform the following steps.
    1. In the **Script** dialog box, click **Create**, and then type a new name for your script.
    2. Select the script you just created and click **Modify**.

3. In the **Script - New Script** dialog box, create your Visual Basic script by typing the information in the text box under **Script**.
4. Click **Test**, and verify that the screen was found
5. Click **OK** to close the **Script - New Script** dialog box.
6. Click **OK** to close the **Scripts** dialog box.
7. Click **OK** to close the **Screen Properties** dialog box.

Use Application Plan Designer Scripts

About using Visual Basic scripts

In Perceptive Content, you can use Microsoft Visual Basic scripting as a tool to extend functionality.

When creating an application plan, you can create, modify, and apply Visual Basic scripts to a property, custom property, or dictionary data element.

> For example, if you have a dictionary element called Full Name with the value "Bruce Jones", you can create a script that reverses the first name and the last name to read "Jones, Bruce".

You can also create and apply a script to the Field1 property to remove all punctuation (that is, dashes) from a social security number. As a result, the Visual Basic script updates the social security number "123-45-6789" to the number "123456789."

When you add a script to an item or container property, the script runs prior to the context mapping. If you select the User Entry source option, however, Perceptive Content clears the script data from the property to allow for user entry of data.

Create a Visual Basic script

In Perceptive Content, you can use Microsoft Visual Basic Scripting as a tool to extend functionality. To create a Visual Basic script, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the appropriate screen, and then do one of the following actions:

Situation	Steps
Create a Visual Basic script for a dictionary data element	1. In the **Dictionary** pane, double-click the dictionary data element for which you want to create a Visual Basic script. 2. In the **Dictionary Field** dialog box, under **Processing**, click **Add > Script**. 3. In the **Script** dialog box, click **Create**, and then type a new name for your script.

Situation	Steps
	4. Select the script you just created and click **Modify**.
Create a Visual Basic script for a map value	1. In the right pane, click the **Map** tab and then double-click the value for which you want to create a Visual Basic script. 2. In the **Attributes** dialog box, under **Advanced**, from the **Scripts** list, select **Manage Scripts**. 3. In the **Script** dialog box, click **Create**, and then type a new name for your script. 4. Select the script you just created and click **Modify**.

2. In the **Script New Script** dialog box, create your Visual Basic script by typing the information in the text box under **Script**.
3. Under **Sample**, in the **Input** box, enter a value that allows you to test the script you just created.
4. Click **Test**, and verify **Perceptive Content** returns the appropriate value.
5. To close the **Script New Script** dialog box, click **OK**.
6. To close the **Scripts** dialog box, click **OK**.
7. To close the **Dictionary Field** dialog box or **Attributes** dialog box, click **OK**.

Apply a Visual Basic script

A Visual Basic script defines how Perceptive Content formats and processes screen data elements before the elements are added to the dictionary, or when applied to a map value. To apply a Visual Basic script, complete the following steps.

- In the Screens pane, select the screen you want, and then do one of the following actions.
  - Apply a Visual Basic script to a dictionary dataelement
  - Apply a Visual Basic script to a map value

Suppose you want to remove the dashes from the social security numbers you map using an application plan in your Human Resources department. To remove the dashes, you apply the following script:

```
workingvalue = DeleteChars(workingvalue, "-")
```

To test this script, in the Input box, type 123-45-6789 and click Test. If the script works properly, "123456789" appears in the Test box. When you apply a Visual Basic script, Perceptive Content runs that script each time a user invokes the application plan.

Modify a Visual Basic script

In Perceptive Content, you can use Microsoft Visual Basic Scripting as a tool to extend functionality. To modify a Visual Basic script, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the appropriate screen, and then do one of the following actions.

Action	Steps
Modify a Visual Basic script for a dictionary data element	1. In the **Dictionary** pane, double-click the dictionary data element for which you want to create a Visual Basic script. 2. In the **Dictionary Field** dialog box, under **Processing**, select the script you want to modify and then click **Modify**. 3. In the **Script** dialog box, select the script you want to modify and then click **Modify**. 4. Select the script you just created and click **Modify**.
Modify a Visual Basic script for a map value	1. In the right pane, click the **Map** tab, and then double-click the value you want to modify in a Visual Basic script. 2. In the **Attributes** dialog box, under **Advanced**, from the **Scripts** list, select **Manage Scripts**. 3. In the **Script** dialog box, select the script you want to modify, and then click **Modify**.

2. In the **Script - New Script** dialog box, modify your Visual Basic script by typing the information in the text box under **Script**.
3. Under **Sample**, in the **Input** box, enter a value that allows you to test the script you just created.
4. Click **Test**, and verify that the script returns the appropriate value.
5. To close the **Script - New Script** dialog box, click **OK**.
6. To close the **Scripts** dialog box, click **OK**.
7. To close the **Dictionary Field** dialog box or **Attributes** dialog box, click **OK**.
8. Close **Application Plan Designer**.

Rename a Visual Basic script

In Perceptive Content, you can use Microsoft Visual Basic Scripting as a tool to extend functionality. To rename a Visual Basic script, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the appropriate screen, and then do one of

the following actions:

Situation	Steps
Rename a Visual Basic script in applied to a dictionary data element	1. In the **Dictionary** pane, double-click the dictionary data element where you want to rename a Visual Basic script. 2. In the **Dictionary Field** dialog box, under **Processing**, select the script you want to rename and then click **Modify**. 3. In the **Scripts** dialog box, select the script you want to rename, and then click **Rename**. 4. Type the new name of the script. 5. To close the **Scripts** dialog box, click **OK**. 6. To close the **Dictionary Field** dialog box, click **OK**.
Rename a Visual Basic script applied to a map value	1. In the right pane, click the **Map** tab, and then double-click the value where you want to rename a Visual Basic script. 2. In the **Attributes** dialog box, under **Advanced**, select **Script > Manage Scripts**. 3. In the **Scripts** dialog box, select the script you want to rename, and then click **Rename**. 4. Type the new name of the script. 5. To close the **Scripts** dialog box, click **OK**. 6. To close the **Attributes** dialog box, click **OK**.

2. Close **Application Plan Designer**.

Remove a Visual Basic script

This procedure removes a visual basic script from a defined data element. To remove a Visual Basic script, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the appropriate screen, and then do one of the following actions:

Situation	Steps
Remove a Visual Basic script applied to a dictionary data element	1. In the **Dictionary** pane, double-click the dictionary data element where you want to remove a Visual Basic script. 2. In the **Dictionary Field** dialog box, under **Processing**, select the script you want to delete and then click **Modify**. 3. In the **Scripts** dialog box, select the script you want to delete and then click **Delete**. 4. In the confirmation dialog box, click **Yes**. 5. To close the **Scripts** dialog box, click **OK**. 6. To close the **Dictionary Field** dialog box, click **OK**.
Remove a Visual Basic script applied to a map value	1. In the right pane, click the **Map** tab, and then double-click the value for which you want to remove a Visual Basic script. 2. In the **Attributes** dialog box, in the **Script** list, select **None**. 3. To close the **Attributes** dialog box, click **OK**.

2. Close **Application Plan Designer**.

Delete a Visual Basic script

To delete a Visual Basic script, complete the following steps.

Deleting a Visual Basic script permanently deletes it from the system for the defined data elements but removing a Visual Basic script only removes it from one data element.

1. In **Application Plan Designer**, select the **Map** tab.
2. Double-click a property, such as Drawer, Folder Name, or Document Name.
3. In the **Script** list, select **Manage Scripts**.
4. In the**Scripts** dialog box, select the Visual Basic script you want to delete.
5. Click **Delete**.
6. In the confirmation dialog box, click **Yes**.
7. Click **OK** until you return to the **Application Plan Designer**.

Examples

Visual Basic script functions

The following list describes key functions you can use in your scripts. Each list item contains information about a function, its syntax, any arguments you need to supply, an example of the function, and the results of the example.

AppGetData

This function returns a screen element.

Syntax

```
String appGetData(int hwnd, String Path, String Type)
```

Arguments

Use `FindWindow` to get the identifier of the window (hwnd). The type can be any of the following quoted strings. Value is the default.

- Name
- Description
- Window
- Value

String `Path` can be a point, the path, or a path/name. When it is a point, it is relative to the top left corner of the host application. The path is designated by children (1|4 would be the first child of the window, and then its fourth child).

A name can be specified using the format `n^String`. It looks for the *Nth* instance of that name, where *N* is the number specified. If the number is not specified it uses the first instance of the name found.

Example

```
Point = appGetData(hwnd, "POINT(150,200)", "Value")
Path = appGetData(hwnd, "4|4|4|3|4|1|1|1|1|1|1|4|1|5|1", "Value")
Name = appGetData(hwnd, "1|2^Test|4|1|1|3", "Value")
```

Result

Result is Smith, John F .

AppGetDataVisible

This function returns a screen element from a visible window.

Syntax

```
String appGetDataVisible(int hwnd, String Path, String Type)
```

Arguments

Use `FindWindow` to get the identifier of the window (hwnd). The type can be any of the following quoted strings. Value is the default.

- Name
- Description
- Window
- Value

String `Path` can be a point, the path, or a path/name. When it is a point, it is relative to the top left corner of the host application. The path is designated by children (1|4 would be the first child of the window, and then its fourth child).

A name can be specified using the format `n^String`. It looks for the *Nth* instance of that name, where *N* is the number specified. If the number is not specified it will use the first instance of the name found.

Example

```
Point = appGetDataVisible(hwnd, "POINT(150,200)", "Value")
```

```
Path = appGetDataVisible(hwnd, "4|4|4|3|4|1|1|1|1|1|1|1|4|1|5|1",
"Value")
```

```
Name = appGetDataVisible(hwnd, "1|2^Test|4|1|1|3", "Value")
```

Result

The result is Smith, John F .

ClipboardGetData

This function returns data from a DOS or terminal application using the clipboard.

Syntax

```
string ClipboardGetData(int hwndSelectedWind, int Position, int Length,
int ConnectionDelay, int ConnectionRetries, string ConnectionString)
```

Arguments

Int `hwndSelectWind` is the window handle of the application plan.

Int `Position` is the starting position. This argument assumes a starting position at 80 columns. For example, 0-80 is line one, 81-161 is line two, 162-242 is line three, and so on.

Int `Length` is the number of characters to capture.

Terminal Capture Arguments

Int `ConnectionDelay` sets the delay in the seconds between connection attempts.

Int `ConnectionRetries` is the number of times to retry.

String `ConnectionString` determines how to capture the data.

Example

```
window = FindWindow("Test App")
results = ClipboardGetData(window, 0, 20, 5, 5, "57634;")
```

Result

The result is Smith, Jones .

DeleteChars

This function deletes all specified characters.

Syntax

```
STR DeleteChars(STR SourceStr, STR CharsStr)
```

Arguments

String `rawValue` is the string from which you want to remove characters.

String `charsToDelete` are the characters to delete from the `rawValue` string.

Example

```
WorkingValue = DeleteChars(folder, "()- ")
```

Result

The string "(913) 111-1111" becomes "9131111111".

DeleteCharsOfType

This function deletes the specified characters of a certain type.

Syntax

```
STR DeleteCharsOfType(STR SourceStr, STR TypeStr)
```

Arguments

p specifies punctuation.

a specifies alphabetical characters.

d specifies numeric characters.

Example

```
WorkingValue = DeleteCharsOfType(folder, "pa")
```

Result

The string abc9999.. becomes 9999 .

DeleteRangeOfChars

This function parses a string and deletes the specified amount of characters from that string.

Syntax

```
STR DeleteRangeOfChars(STR SourceStr, int StartPos, int Len)
```

Arguments

String `rawValue` is the string from which to remove characters.

Integer `startChar` is the starting position in the `rawValue` string.

Integer `length` is the number of characters after `startChar` to delete.

Example

```
WorkingValue = DeleteRangeOfChars(folder, 0, 3)
```

Result

The string abcdefgh becomes defgh .

DoMacroAlpha

This function removes all non-alphabetic characters.

Syntax

```
STR DoMacroAlpha(STR rawValue)
```

Arguments

String `rawValue` is the string of non-alphabetic characters you want to remove.

Example

```
msgbox "Alpha: " + DoMacroAlpha("a1@b2#c3$")
```

Result

The string a1@b2#c3$ becomes abc .

DoMacroAlphaNumeric

This function removes all non-alphanumeric characters.

Syntax

```
STR(STR rawValue)
```

Arguments

String `rawValue` is the string from which to remove non-alphanumeric characters.

Example

```
msgbox "AlphaNumeric: " + DoMacroAlphaNumeric("a1@b2#c3$")
```

Result

The string "a1@b2#c3$" becomes a1b2c3 .

DoMacroDateConversion

This function converts a string variable to a variable of date type.

Syntax

```
STR(STR rawValue, string dateFormat)
```

Arguments

String `rawValue` is the string to convert to a date.

String `dateFormat` is the format in which the date string was passed. You can include the following format variables:

- m represents a month number
- M represents a month letter
- d represents day
- y represents year

Example

```
msgbox "Date: " + DoMacroDateConversion("OCT 16th, 1978", "MMM dd, yyyy")
```

Result

The string OCT 16th, 1978 becomes 10/16/1978 .

DoMacroDeleteCharacters

This function removes specific characters from a string.

Syntax

```
STR(STR rawValue", STR charsToDelete)
```

Arguments

String `rawValue` is the value from which to remove the characters.

String `charsToDelete` is the list of characters to remove from rawValue.

Example

```
msgbox "Delete Characters: " + DoMacroDeleteCharacters("INV-778989",
"INV-")
```

Result

The string INV-778989 becomes 778989 .

DoMacroLeadingZeros

This function removes leading zeros from a string.

Syntax

```
STR(STR or int rawValue)
```

Arguments

String `rawValue` is the value from which the leading zeros are removed.

Example

```
msgbox "Remove Zeros: " + DoMacroLeadingZeroes("000012345")
```

Result

The string 000012345 becomes 12345 .

DoMacroFixedLength

This function returns the designated amount of characters from a string.

Syntax

```
STR DoMacroFixedLength(STR rawValue, int length, bool fromLeft)
```

Arguments

String `rawValue` is the value to convert.

Integer `length` is the number of characters to return from the string value.

If the boolean `fromLeft` is set to true, the substring returned is the number of characters starting from the left, plus the value length. If `fromLeft` is set to false, the substring returned is the number of characters starting from the right, plus the value length.

Example

```
msgbox "Fixed Length (left): " + DoMacroFixedLength("abcdefg", 4, true)
```

Result

The string abcdefg returns abcd .

DoMacroNumberConversion

This function returns a value of type number.

Syntax

```
STR(STR or int rawNum, int numOfDecimals, char separator)
```

Arguments

String or integer `rawNum` is the value to format as a number.

Integer `numOfDecimals` is the total decimal numbers.

Character `separator` is the decimal separator to use.

Example

```
msgbox "Number: " + DoMacroNumberConversion("1234.12", 0, ".")
```

Result

The string 1234.12 becomes 1234 .

DoMacroNumeric

This function converts a string variable to a numeric variable.

Syntax

```
STR(STR rawValue)
```

Arguments

String `rawValue` is the string from which to remove non-numeric characters.

Example

```
msgbox "Number: " + DoMacroNumber("INV-8899878")
```

Result

The string INV-8899878 returns 8899878 .

DoMacroLineSearch

This function searches multiple lines based on the input string and extracts data from the specified line.

Syntax

```
STR DoMacroLineSearch(STR Input, int LineNum, STR Direction, int
StartPos, STR StopType, STR StopValue)
```

Arguments

String `Input` is the string to search.

Integer `LineNum` is the line of text from input to process (1 is the first line).

String `Direction` is either "Forward" or "Backward" and is the direction to search.

Integer `StartPos` is where in the line of text the data starts. When direction is backward, the first character position is the end of the line.

String `StopType` indicates how many characters from the StartPos to return. Valid quoted string values include:

- End Of Line
- Character Length
- Stopping Characters

Example

```
input = "this is a test" & vbNewLine & "This is only a test" &
vbNewLine & "Testing 1, 2, 3"

text1 = DoMacroLineSearch(input, 2, "Forward", "9", "Character
Length", "6")
```

Result

The result string is only a .

Additional information

If StopType is "Character Length", StopValue is a number of characters to return. If StopType is "Stopping Characters", the StopValue is a character pattern that indicates where to stop the string returned.

DoMacroStaticSearch

This function searches for text based on the input string and returns text from the results.

Syntax

```
STR DoMacroStaticSearch(STR Input, STR AnchorText, int LineOffset, STR
Direction, STR StopType, STR StopValue)
```

Arguments

String `Input` is the string to search.

String `AnchorText` is the text to search for.

Integer `LineOffset` is the number of lines above or below the anchor text to select. Zero(0) selects the line that contains the `AnchorText`, a negative number s a line above and a positive number selects a line below.

String `Direction` is either "Forward" or "Backward" and is the direction to search.

String `StopType` indicates how many characters from the StartPos to return. Valid quoted string values include:

- End Of Line
- Character Length
- Stopping Characters

Example

```
Input = "this is a test" & vbNewLine & "This is only a test" &
vbNewLine & "Testing 1, 2, 3"

"Forward", "End Of Line",)
```

Result

The result string is Testing 1, 2, 3 .

Additional information

If StopType is "Character Length", StopValue is a number of characters to return. If StopType is "Stopping Characters", the StopValue is a character pattern that indicates where to stop the string returned.

DoMacroUser

This function returns a value of type Perceptive Content username.

Syntax

```
DoMacroUser(String userName)
```

Arguments

String `userName` is the user name to pass into Perceptive Content.

Example

```
msgbox "User: " + DoMacroUser("jsmith")
```

Result

The result is User:jsmith .

DocProperty

This function can set a property for a document or return the desired value.

Syntax

```
string DocProperty(string customPropName)
```

Arguments

String `customProperty` is the name of the custom property that needs to be set.

Example

DocProperty("Name") = " Jane Doe"

Result

The result string is Name: Jane Doe .

Field

This function can set or return the current value from an element in the dictionary.

Syntax

```
string Field(string dictionaryElement)
```

Arguments

String `dictionaryElement` is the dictionary element name.

Example

```
msgbox Field("Last Name")
```

Result

Returns the current value from the dictionary element called Last Name.

FolderProperty

This function can set a property for a folder or return a specific value.

Syntax

```
string FolderProperty(string customPropName)
```

Arguments

String `customPropName` is the name of the custom property that needs to be set.

Example

```
FolderProperty("Patient ID") = "9910552"
```

Result

The result string is Patient ID: 9910552 .

FindWindow

This function searches the desktop for a window title that matches the argument.

Syntax

```
int FindWindow(STR WindowTitle)
```

Arguments

String `WindowTitle` is the title to search for and can include wildcards (* or ?)

Example

```
hwnd = FindWindow("Internet Explorer")
```

Result

The result is an integer that identifies the window in the system.

Additional information

The returned value is most useful when used as the first parameter to `AppGetData`.

GetActiveScreenName

This function returns the user defined screen name that Perceptive Content is currently using.

Syntax

```
GetActiveScreenName()
```

Example

```
msgbox GetActiveScreenName()
```

Result

Returns the current user defined screen name.

GetINowINIString

This function reads a string from the *ImageNow.ini* file.

Syntax

```
string GetINowINIString(string SectionName, string KeyName)
```

Arguments

String `SectionName` is the name of the section in the INI file to search for the key.

Example

```
Result = GetINowINIString("Logon Settings", "Default")
```

Result

Result contains the name of the default Perceptive Content connection profile.

GetScreenText

This function gets the text on the screen at the supplied x and y coordinates, to the specified length.

Syntax

```
String GetScreenText(int Xpos, int Ypos, int Len)
```

Arguments

Integer `Xpos` is the horizontal position of the first character retrieved with a value of 1 indicating the leftmost character.

Integer `Ypos` is the vertical position of the first character retrieved with a value of 0 being the topmost line.

Integer `Length` is the number of characters to return.

Example

```
WorkingValue = GetScreenText(0, 8, 15)
```

Result

The result string is City: Crestview .

HyperlearnCreateFont

This function creates the font string necessary for HyperlearnGetData.

Syntax

```
string HyperlearnCreateFont(string fontName, int fontSize, int Weight,
int Italic)
```

Arguments

String `fontName` is the name of the font.

Integer `fontSize` is the size of the font.

Integer `Weight` is 400 for normal weight fonts, 700 for bold.

Integer `Italic` is 0 for normal, 1 for italic.

Example

```
font = HyperlearnCreateFont("Courier", 12, 400, 0)
```

Result

The result string is -11,0,0,0,400,0,0,0,0,0,0,0,0,Courier .

HyperlearnGetData

This function extracts the text from an area of a window using Hyperlearn technology.

Syntax

string HyperlearnGetData(int selectedWindow, int left, int top, int width, int height, string font)

Arguments

Integer `hwndSelectWind` is the window handle from which to get data. Use the `FindWindow()` method to get the window handle.

Integer `left` is the left side of the rectangle that specifies where HyperLearn should look for text.

Integer `top` is the top of the rectangle that specifies where HyperLearn should look for text.

Integer `height` is the height of the rectangle that specifies where HyperLearn should look for text.

String `font` specifies the font HyperLearn should use when looking for text in the specified rectangle.

Example

```
hwnd = FindWindow("PSI higher Ed Demo")
font = HyperlearnCreateFont(Courier, 12, 400, 0)
result = HyperlearnGetData(hwnd, 100, 200, 50 16 font)
```

Result

The text extracted from the area at pixel coordinates 100, 200 with a width of 50 pixels and a height of 16 pixels from the window titled "PSI Higher Ed Demo."

Additional information

HyperlearnCreateFont creates the font string. You must first add fonts to the application plan to receive results.

IntToStr

This function converts an integer to a string.

Syntax

```
String IntToString(int Num)
```

Arguments

Integer `Num` is the integer value to convert to a string format.

Example

```
s=IntToString(99)
```

Result

The result string is 99 .

Additional information

This method was deprecated in version 6.7. Replace it with the Visual Basic Scripting method, CStr.

LogError

This function writes a message to the client log file.

Syntax

```
void LogError(String message)
```

Arguments

Message to be written to the log

Example

```
LogError( an error occurred in the LearnMode script )
```

Result

The result string is added to the client log file.

LogWarning

This function writes a message to the client log file.

Syntax

```
void LogWarning(String message)
```

Argument

```
Message to be written to the log
```

Example

```
LogWarning( something happened in the LearnMode script )
```

Result

If the client log level is 1 or greater, the string is added to the client log file. If the client log level is 0, then no action is taken.

NameReorder

This function converts a string that contains a first name, a space, and a last name to a last name, comma, first name format.

Syntax

```
STR NameReorder(STR SourceStr)
```

Arguments

String `SourceStr` is the string that is reordered.

Example

```
WorkingValue = NameReorder(folder)
```

Result

The string John Doe becomes Doe, John .

PostDocProperty

This function can set a property for a folder or return the desired value.

Syntax

```
string PostDocProperty(string customPropName)
```

Arguments

String `customPropName` is the name of the custom property that needs to be set.

Example

```
PostDocProperty("Patient ID") = "9910552"
```

Result

The string result is 9910552 .

PostField

This function can set the current value or return from an element in the dictionary.

Syntax

```
string PostField(string dictionaryElement)
```

Arguments

String `dictionaryElement` is the dictionary element name.

Example

```
msgbox PostField("Last Name")
```

Result

Returns the current value from the dictionary element called Last Name.

PostFolderProperty

This function can set a property for a folder or return a specific value.

Syntax

```
string PostFolderProperty(string customPropName)
```

Arguments

String `customPropName` is the name of the custom property that needs to be set.

Example

```
PostFolderProperty("Name") = " Joe Smith"
```

Result

The result string is Name: Joe Smith .

SetFocus

Searches for a named window and gives it input focus.

Syntax

```
int SetFocus(string windowTitle)
```

Arguments

String `windowTitle` is the title of the window for which to search.

Example

```
success = SetFocus("PSI Higher Ed Demo")
```

Result

The window titled "PSI Higher Ed Demo" moves to the foreground and has input focus. Success equals 1 if the window is found.

SetForeground

This function finds the window with a given handle and gives it input focus.

Syntax

```
int SetForeground(Int64 hWnd)
```

Arguments

Integer `hWnd` is the window handle that receives focus.

Example

```
hwnd = FindWindow("PSI Higher Ed Demo")
result = SetForeground(hwnd)
```

Result

The window titled "PSI Higher Ed Demo"moves to the foreground and has input focus. Success equals 1 if the window is found.

ShellOpen

This function launches a program or a file.

Syntax

```
ShellOpen(String Command)
```

Arguments

Command is a program or a file to open.

Example

```
ShellOpen("c:\Somefile.txt")
ShellOpen("calc.exe")
```

Result

NotePad opens with *c:\Somefile.txt* loaded. The calculator application also is opened.

Sleep

This function establishes the amount of time to pause without doing anything.

Syntax

```
void Sleep(int msec)
```

Arguments

Supply the time for `msec` in milliseconds.

Example

```
Sleep(100) ' 1/10th of a second
```

Result

The script is paused for 1/10th of a second.

SwipSwapFromBegin

This function takes a specified number of characters from the beginning of a string and moves them to the end of the string.

Syntax

```
STR SwipSwapFromBegin(STR SourceStr, int Len)
```

Argument

Supply the string in `SourceStr`. Integer `Len` is the number of characters to take, starting at the beginning of the string.

Example

```
WorkingValue = SwipSwapFromBegin(folder, 3)
```

Result

The string result is abcdefgh becomes defghabc .

SwipSwapFromEnd

This function takes a specified number of characters from the end of a string and moves them to the beginning of the string.

Syntax

```
STR SwipSwapFromEnd(STR SourceStr, int Len)
```

Arguments

Supply the string in `SourceStr`. Integer `Len` is the number of characters to take, starting at the end of the string.

Example

```
WorkingValue = SwipSwapFromEnd(folder, 3)
```

Result

The result string abcdefgh becomes fghabcde .

SwipSwapFromPos

This function moves a specified number of characters from one position to another in a string.

Syntax

```
STR SwipSwapFromPos(STR SourceStr, int StartPos, int Len, int EndPos)
```

Arguments

Supply the string in `SourceStr`. Integer

`StartPos` is the starting character position beginning at 0.

Integer `Len` is the number of characters to take.

Integer `Endpos` is the character position where the string is moved, where the first character is 0.

Example

```
WorkingValue = SwipSwapFromPos(folder, 4, 2, 1)
```

Result

The string abcdefgh becomes aefbcdgh .

Term

This function searches for windows to close based on "CLASS" or "TITLE."

Syntax

```
int Term String Mode, String Name, String Command
```

Arguments

String `Mode` specifies how to search for windows to close based on "CLASS", or "TITLE." "CLASS" searches for windows with a class that exactly matches the Name parameter. "TITLE" searches for windows whose title text contains the Name parameter.

String `Name` is the string to match in the window title or class. This parameter is case sensitive.

String `Command` specifies how to close matched windows and can be "QUIT," or "CLOSE." In general, windows that receive the "CLOSE" message prompt the user to save changes before closing. The "QUIT" message closes the application immediately.

Example

```
Term = "TITLE," "NotePad," "QUIT."
```

Result

All open NotePad windows are closed.

TopMsg

This function displays a window with the specified text.

Syntax

```
TopMsg(string message)
```

Arguments

String `message` is the message to display.

Example

```
TopMsg = "Hello World"
```

Result

A dialog box displays the text "Hello World" inside.

WindowTitle

This function returns a specified number of characters in the window title of the current window.

Example

```
myWindowTitle = WindowTitle
```

Result

The string result is PSI Banking Demo .

WinGetData

This function returns text from a window using WindowWalker.

Syntax

```
string WinGetData(int windowHandle, string directions)
```

Arguments

Integer `selectedWindow` is the handle to the window from which to get data. Use the `FindWindow()` function to get the window handle.

String `directions` is the information passed to the WindowWalker extractor to access the control.

Example

```
hwnd = FindWindow("PSI Higher Ed Demo")
result = WinGetData(hwnd, "D3;L7")
```

Result

The result string is 206528888 .

Testing VB script functions

The Test button in the script editor does not always produce valid results when using these functions. To test your code, save the template, close the application plan, and then click on the  icon in the Perceptive Content toolbar to view the results.

Visual Basic script variables

Below is a list of variables you can use in your Visual Basic scripts.

The Test button in the script editor does not always produce valid results when using these functions. To test your code, try saving the template, closing the application plan, and then clicking on the icon in the ImageNow Client toolbar to view the results.

Note: In previous versions of Perceptive Content, you used Folder and Tab. In Perceptive Content 6.7, these supported document key variables changed to Field1 and Field2. While Folder and Tab are currently supported, they may be deprecated in future versions.

Document key variables

Drawer

DocName

Field1

Field2

Field3

Field4

Field5

DocType

Post-macro document key variables

PostDrawer

PostDocName

PostField1

PostField2

PostField3

PostField4

PostField5

PostDocType

Other variables

User - Logged in user name

Group - Logged in user group name

AppPlan - Name of the current application

Reserved variables

WorkingValue - Sets the current dictionary element to the desired value. This can also be used to get the current dictionary element.

ScreenFound - The default for this value is FALSE. The script sets the value to TRUE if the screen is found. If FALSE, the next available screen is used.

Visual Basic scripts for DDE

You can use the following functions in any script in application plan types. These functions are useful with the DDE LearnMode type.

The Test button in the script editor does not always produce valid results when using these functions. To test your code, try saving the template, closing the application plan, and then clicking on the icon in the ImageNow Client toolbar to view the results.

Note: These Dynamic Data Exchange methods are deprecated in Perceptive Content 6.7. In previous versions of Perceptive Content, you used Folder and Tab. In Perceptive Content 6.7, these supported document key variables changed to Field1 and Field2. While Folder and Tab are currently supported, they may be deprecated in future versions.

Script Name	Syntax
DDEExecute	int DDEExecute(int Channel, STR CommandStr)
DDEInitiate	int Channel = DDEInitiate(STR AppStr, STR TopicStr)
DDEPoke	int DDEPoke(int Channel, STR ItemStr, STR DataStr)
DDERequest	STR DDERequest(int Channel, STR ItemStr)
DDETerminate	int DDETerminate(int Channel)

DDE Script Example

The following example script reads keys from Microsoft Excel.

```
nChannel = DDEInitiate("excel", "test.xls")
if nChannel <> -1 then
s = DDERequest(nChannel, "R1C1")
field1 = left(s, len(s)-2)
s = DDERequest(nChannel, "R1C2")
field2 = left(s, len(s)-2)
s = DDERequest(nChannel, "R1C3")
field3 = left(s, len(s)-2)
s = DDERequest(nChannel, "R1C4")
field4 = left(s, len(s)-2)
s = DDERequest(nChannel, "R1C5")
field5 = left(s, len(s)-2)
DDETerminate(nChannel)
end if
```

Visual Basic scripts for Internet Explorer

You can use the functions described in this section in any script in application plan types. These functions are only effective in capturing values from Microsoft Internet Explorer pages. They are useful with the Internet Explorer LearnMode type.

IECapture

This function captures the data from Internet Explorer in the form of name/value pairs and stores them for subsequent searches. It returns the number of pairs found.

Syntax

```
IECapture (bool useTDtags, HWND hwndSelectedWind)
```

Arguments

`useTDTags`: use this boolean if LTD tags need to be captured.

`hwndSelectedWind`: specifies the handle for the window to capture the data from, 0 to use the top-most Internet Explorer window.

Example

```
size = IECapture(false, 0)
```

Results

0: <PEB#1>,"Student Name:"

1: <PEB#2>,"Smith, James"

2: <PEB#3>, "Student ID:",

3: <PEB#4>,"1234567"

4: <PEB#5>,"Date of Birth:"

5: <PEB#6>, "6/14/1987"

size = 6

Example

```
size = IECapture(true, 0)
```

Results

0: <LTD#1>, Student Name:

1: <LTD#2>, Smith, James

2: <LTD#3>, Student ID:

3: <LTD#4>, 1234567

4: <LTD#5>, Date Of Birth:

5: <LTD#6>, 6/14/1987

6: <LTD#7>, First Term:

7: <LTD#8>, Fall 2008

8: <PEB#1>, Student Name:

9: <PEB#2>, Smith, James

10: <PEB#3>, Student ID:

11: <PEB#4>, 1234567

12: <PEB#5>, Date of Birth:

13: <PEB#6>, 6/14/1987

size = 14

Note: You must use the IECapture or the IECaptureEx function first to store data for subsequent searches. All the following function examples use IECapture(true, 0) or IECaptureEx(true, 0, myArray).

IECaptureEx

This function captures a specified set of named fields from Internet Explorer and stores them for subsequent searches.

Syntax

```
IECaptureEx (bool useTDtags, HWND hwndSelectedWind, VARIANT* pVarIDs)
```

Arguments

`useTDTags`: use this boolean if LTD tags need to be captured.

`hwndSelectedWind`: specifies the handle for the window to capture the data from, 0 to use the top-most Internet Explorer window.

`pVarIDs`: a list of IDs to capture from Internet Explorer.

Example

```
myArray = Array( CSID ,  CSID2
size = IECaptureEx(false, 0, myArray)
```

Results

0: CSID, 12n53n43

1: CSID2, 25n232

2: <PEB#1>, Student Name:

3: <PEB#2>, Smith, James

4: <PEB#3>, Student ID:

5: < PEB#4>, 1234567

6: <PEB#5>, Date of Birth:

7: <PEB#6>, 6/14/1987

size = 8

Example

```
myArray = Array(  )
size = IECaptureEx(true, 0, myArray)
```

Results

0: <LTD#1>, Student Name:

1: <LTD#2>, Smith, James

2: <LTD#3>, Student ID:

3: <LTD#4>, 1234567

4: <LTD#5>, Date of Birth

5: <LTD#6>, 6/14/1987

6: <LTD#7>, First Term:

7: <LTD#8>, Fall 2008

8: <PEB#1>, Student Name:

9: <PEB#2>, Smith, James

10: <PEB#3>, Student ID:

11: <PEB#4>, 1234567

12: < PEB#5>, Date of Birth

13: <PEB#6>, 6/14/1987

size = 14

IEGetNameByIndex

This function returns the name at the specified index.

Syntax

```
IEGetNameByIndex (int nIndex)
```

Arguments

`nIndex`: specifies the location of the tag. Zero returns the first tag.

Example

```
name = IEGetNameByIndex(1)
```

Results

name = <LTD#2>

Note: You must use the IECapture function first to store data for subsequent searches. This example uses IECapture(true, 0).

IEGetValueByIndex

This function returns the value at the specified index.

Syntax

```
IEGetValueByIndex (int nIndex)
```

Arguments

`nIndex`: specifies the location of the value. Zero returns the first value.

Example

```
value = IEGetValueByIndex(1)
```

Results

value = Smith, James

Note: You must use the IECapture function first to store data for subsequent searches. This example uses IECapture(true,0).

IEFindName

This function allows you to search for a name and it returns the location where it was found. It returns -1 if not found.

Syntax

```
IEFindName (LPCTSTR name, int nStart, bool bCaseSensitive)
```

Arguments

`Name`: specifies the name to search for.

`nStart`: specifies the location to start at. Use 0 to start the search from the beginning

`bCaseSensitive`: use this boolean if the search is case sensitive.

Example

```
nameLocation = IEFindName ( <PEB#1> , 0, false)
```

Results

nameLocation = 8

Note: This function supports wild card characters such as * and ?. You must use the IECapture function first to store data for subsequent searches. This example uses IECapture(true,0).

IEFindNameReverse

This function allows you to search for a name in reverse and it returns the location where it was found. It returns -1 if not found.

Syntax

```
IEFindNameReverse (LPCTSTR name, int nStart, bool bCaseSensitive)
```

Arguments

`Name`: specifies the name to search for.

`nStart`: specifies the location to start at. Use -1 to start at the end.

`bCaseSensitive`: use this boolean if the search is case sensitive.

Example

```
nameLocation = IEFindNameReverse ( <PEB#1> , 6, true)
```

Results

nameLocation = -1

Note: This function supports wild card characters such as * and ?. You must use the IECapture function first to store data for subsequent searches. This examples uses IECapture(true,0).

IEFindValue

This function allows you to search for a value and it returns the location where it was found. It returns -1 if not found.

Syntax

```
IEFindValue (LPCTSTR value, int nStart, bool bCaseSensitive)
```

Arguments

`Value`: specifies the value to search for.

`nStart`: specifies the location to start at. Use 0 to search from the beginning.

`bCaseSensitive`: use this boolean if the search is case sensitive.

Example

```
valuelocation = IEFindValue( Student Name: , 0, true)
```

Results

valueLocation = 0

Note: This function supports wild card characters such as * and ?. You must use the IECapture function first to store data for subsequent searches. This example uses IECapture(true,0).

IEFindValueReverse

This function allows you to search for a value in reverse and it returns the location where it was found. It returns -1 if not found.

Syntax

```
IEFindValueReverse (LPCTSTR value, int nStart, bool bCaseSensitive)
```

Arguments

`Name`: specifies the value to search for.

`nStart`: specifies the location to start at. Use -1 to start at the end.

`bCaseSensitive`: use this boolean if the search is case sensitive.

Example

```
valueLocation = IEFindValueReverse( Student Name: , -1, true)
```

Results

valueLocation = 8

Note: This function supports wild card characters such as * and ?. You must use the IECapture function first to store data for subsequent searches. This example uses IECapture(true,0).

IEGetValue

This function returns the value when passed the name of the name/value pair.

Syntax

```
IEGetValue (LPCTSTR value, bool bCaseSensitive)
```

Arguments

`Name`: specifies the name to search for.

`bCaseSensitive`: use this boolean if the search is case sensitive.

Example

```
WorkingValue = IEGetValue ( <LTD#12> )
```

Results

The value, which corresponds to the name <LTD#12>, is placed into folder.

Note: This function supports wild card characters such as * and ?. You can use the IEGetValue function as a shortcut to calling IECapture and then IEGetValueByName. However, a single IECapture and multiple IEGetValueByName calls is a faster option if you need multiple values.

IEGetValueByName

This function allows you to search for a name and it returns the corresponding value. It returns an empty string if not found.

Syntax

```
IEGetValueByName (LPCTSTR name, bool bCaseSensitive)
```

Arguments

`Name`: specifies the name to search for.

`bCaseSensitive`: use this boolean if the search is case sensitive.

Example

```
value = IEGetValueByName( <PEB#1> , false)
```

Results

value = Student Name:

Note: This function supports wild card characters such as * and ?. You must use the IECapture function first to store data for subsequent searches. This example uses IECapture(true,0).

The Test button in the script editor does not always return valid results when you use these functions. To test your code, save the template, close the application plan, and then click the icon in the ImageNow Client toolbar to view the results.

Use macros

Add a macro

A macro is a script you add to an application plan to define how Perceptive Content formats and processes data elements before the elements are added to the dictionary. To add a macro, complete any of the following procedure.

- Add an alphanumeric macro.
- Add a concatenation macro
- Add a date conversion macro
- Add a delete characters macro
- Add a fixed length macro
- Add a line search macro

- Add a number format macro
- Add a split attribute at delimiter macro
- Add a static search macro
- Add a trim characters macro

Macro definitions

The following predefined macros are available for data processing.

Macro Name	Description
Alpha Numeric - Alpha Characters Only	Removes non-alphabetic characters from the output string.
Alpha Numeric - Numeric Characters Only	Removes non-numeric characters from the output string.
Alpha Numeric - Alphanumeric Characters Only	Removes non-alphanumeric characters from the output string. This includes all symbols and punctuation marks.
Concatenation	Joins dictionary data elements.
Date Conversion	Parses the value from you business application to allow for custom properties to accept it. If you want to assign a value to a date custom property, you need to select this macro option to parse the date from your business application to use the format of year, month, and day (yyyy-MM-dd).
Delete Characters	Removes specific characters from the output string.
Fixed Length	Limits the number of characters and digits in the output string.
Line Search	Captures a value on a specific line.
Number Format	Displays a number string in a specific format.
Split at Delimiter	Captures a value in a string that is separated by a delimiter character.
Static Search	Captures a value based on a specific string.
Trim Character	Removes characters from the output string.

Patterns for formatting macros

When creating a macro, you can use the below elements to construct a string to convert the date from your business application into the format of year, month, and day. The Date Conversion macro converts the value from your business application into a format that custom properties can accept.

Format types for date conversion

The format types "d" and "y" must be lowercase and the letter "M" must be uppercase. All other text entered in the Format field is treated literally, such as a forward slash for separating the day, month, and year.

Format Type	Definition	Example
d	Day of the month as digits without leading zeros for single-digit days	For January 6, 2011, d = 6
dd	Day of the month as digits with leading zeros for single-digit days	For January 6, 2011, dd = 06
M	Month as digits without leading zeros for single-digit months	For January 6, 2011, M = 1
MM	Month as digits with leading zeros for single-digit months	For January 6, 2011, MM = 01
MMM	Abbreviated text for month	For January 6, 2011, MMM = Jan
MMMM	Full text for month	For January 6, 2011, MMMM = January
yy	Year represented only by the last two digits	For January 6, 2011, yy = 11
yyyy	Year represented by four digits	For January 6, 2011, yyyy = 2011

Examples

String from host application	Format string to parse
Jan 15, 1962	MMM d, yyyy
15/01/1962	dd/MM/yyyy

Use sequences

What is a sequence?

You can define your own sequence number to use as a document or record property.

When selected, Perceptive Content creates a new set of numbers for each document or record based on the settings you define for the width and increment.

For example, you can create a sequence called Sequence_docproperty1 where the width is 5 and the increment is 1; the first value will be 00001, the second value is 00002, and so forth.

Create a sequence

You can define a sequence in your application plan map and use it as an item property value. To create a sequence, complete the following steps.

1. In **Application Plan Designer**, in the **Screens** pane, select the screen with map values that you want to modify.
2. In the right pane, click the **Map** tab and perform one of the following actions.

Situation	Steps
Under **Document Map**, for the **Document Level**	• Double-click **Document Name**, **Field1**, **Field2**, **Field3**, **Field4**, or **Field5**.
Under **Document Map**, for the **Folder Level**	• Double-click **Folder Name**.
Under **Shortcut Map**, for the **Folder Level**	• Double-click **Folder Name**.
Under **Record Map**, for the **Record Level**	• Double-click **Name**, **Field1**, **Field2**, **Field3**, **Field4**, **Field5**, **Author**, **Originating Organization**, **Location**, **Addressees**, **Other Addressees**, **Media Type**, or **Format**.
Under **Record Map**, for the **Record Folder Level**	• Double-click **Record Folder Name** or **Record Folder Location**.

3. In the **<Property> Attributes** dialog box, under **General**, in the **Source** list, select **Sequence Number**.
4. In the **Value** list, select **Edit Sequence**.
5. In the **Sequence Number** dialog box, click **New**.
6. In the **New Sequence** dialog box, complete the following substeps.
    1. In the **Name** box, type a name for your sequence.

2. From the **Base** list, select the numeral system you want to use for your sequence.
3. In the **Width** box, type the number of digits you want to use in your sequence.
4. In the **Increment** box, type the amount that you want added to your sequence.
5. In the **Prefix** box, type a static text string that you want to precede the sequence.
6. In the **Suffix** box, type a static text string that you want to follow the sequence.

7. Click **OK** to close the **New Sequence** dialog box.
8. Click **Close** to close the **Sequence Number** dialog box.
9. Click **OK** to close the **<Property> Attributes** dialog box.

Modify a sequence

You can define sequence numbers in your Document or Folder map and use them as document property values. To modify a sequence, complete the following steps.

1. In **Management Console**, in the **Screens** pane, select the screen for which you want to modify a sequence.
2. In the right pane, click the **Map** tab and perform one of the following actions depending on your map type:
   - Document Map
     1. Double-click the **Document Name**, **Field1**, **Field2**, **Field3**, **Field4**, or **Field5** property.
     2. In the **<Property> Attributes** dialog box, under **General**, select **Sequence Number** from the **Source** list and then from the **Value** list select **Edit Sequence**.
   - Folder Map
     1. Double-click the **Folder Name** property.
     2. In the **Folder Name Attributes** dialog box, under **General**, select **Sequence Number** from the **Source** list and then from the **Value** list select **Edit Sequence**.
3. In the **Sequence Number** dialog box, select the sequence you want to modify and click **Modify**.
4. In the **Edit Sequence** dialog box, do the following substeps:
   1. In the **Name** box, type a name for your sequence.
   2. From the **Base** list, select the numeral system to use for your sequence.
   3. In the **Width** box, type the number of digits to use in your sequence.
   4. In the **Increment** box, type the amount to add to your sequence.
   5. In the **Prefix** box, type a static text string to precede the sequence.
   6. In the **Suffix** box, type a static text string to follow the sequence.
5. To close the **Sequence Number** dialog box, click **Close**.
6. To close the **<Property> Attributes** dialog box, click **OK**.
7. Close **Application Plan Designer**.

Delete a sequence

To delete a sequence, complete the following steps.

1. In **Application Plan Designer**, click the **Map** tab.
2. In the **Screens** pane, select the screen for which you want to delete the sequence.
3. Double-click on of the following properties:
   - For folder mapping, select **Folder Name.**
   - For document mapping, select **Document Name**, **Field1**, **Field2**, **Field3**, **Field4**, or **Field5**.
4. In the **<Property> Attributes** dialog box, in the **Source** list, select **Sequence Number**.
5. In the **Value** list, select**Edit Sequence.**
6. In the **Sequence Number**dialog box, select the sequence you want to delete.
7. Click **Delete**.
8. In the confirmation dialog box, click **Yes**.
9. Click **Close** and **OK** to return to the **Application Plan Designer**.

Rename a sequence

You can define sequence numbers in your Document or Folder map and use them as document property values. To rename a sequence, complete the following steps.

1. In **Application Plan Designer**, click the **Map** tab.
2. In the **Screens** pane, select the screen for which you want to enable mapping.
3. Do one of the following actions:
   - Document Map
     1. Double-click the **Field1**, **Field2**, **Field3**, **Field4**, or **Field5** property.
     2. In the **<Property> Attributes** dialog box, under **General**, select **Sequence Number** from the **Source** list and then from the **Value** list select **Edit Sequence**.
   - Folder Map
     1. Double-click the **Folder Name** property.
     2. In the **Folder Name Attributes** dialog box, under **General**, select **Sequence Number** from the **Source** list and then from the **Value** list select **Edit Sequence**.
4. In the **Sequence Number** dialog box, select the sequence you want to rename and click **Rename**
5. Type the new name of your sequence.
6. Click **Close** to close the **Sequence Number** dialog box.
7. Click **OK** to close the **<Property> Attributes** dialog box.
8. Close **Application Plan Designer**.