
Hyland®

Training

appGetTree Learnmode

Lab Workbook

Hyland Software, Inc.
28500 Clemens Road
Westlake, Ohio 44145

P: (440) 788-5000
F: (440) 788-5100
Training.Hyland.com

OnBase
© Hyland Software, Inc.
All rights reserved.
v. 2023.02.22CA1100 rg

Information in this document is subject to change without notice and does not represent a commitment on the part of Hyland Software, Inc. or any of its affiliates. The information provided in this document is provided to you under a license agreement or nondisclosure agreement and may be used or disclosed only according to the terms such agreement. Any other use or disclosure is strictly prohibited. All data, names, and formats used in this document's examples are fictitious unless noted otherwise. This document may reference websites operated by third parties. In such case, Hyland has no control or liability for the content of such third party websites. The inclusion of such link shall not constitute an endorsement or affiliation with such third party website; the reference is provided for information purposes only. Should you have any questions pertaining to discrepancies in this document, please contact Hyland Software, Inc. The software described in this document is furnished only under a license agreement and may be used only in accordance with the terms of such agreement. It is against the law to copy the software except as specifically allowed in the license agreement or without the expressed written consent of Hyland Software, Inc. Any unauthorized duplication or use of the software or its corresponding documentation is forbidden.

OnBase, Hyland, Hyland Software, ShareBase, Brainware, Acuo, Nilread, Pacsgear, ShareBase, Saperion, and Perceptive Content are registered or unregistered trademarks of Hyland Software, Inc. or its affiliates in the United States and many other countries. All other products names are registered trademarks or trademarks of their respective companies.

CONTENTS

appGetTree Learnmode	1
Section 1: appGetTree.....	5
appGetTree Configuration	6
appGetTree Overview	6
Launch the business application in chrome.....	6
Navigate to learnmode application plans	6
Create the Application Plan	6
LEarnmode Application plan Designer.....	7
Configure and implement the appgettree script to create the debOut file	7
Locating screen data on the debout file and Script Config to grab value.....	8
Configuring Appgettree for Field1	8
Configuring AppGetTree for field2.....	9
Configuring the appgettree script for The remaining fields	10
Update application Plan Document Key labels.....	11
Test the application plan.....	11
Section 2: appGetTree with Folders.....	12
appGetTree Configuration for Folders.....	13
appgettree with folders overview.....	13
Launch the business application in chrome.....	13
Create the Application Plan	13
LEarnmode Application plan Designer.....	13
Create Dictionary Values.....	13
MapPing Dictionary Values	14
Create a Shortcut Map.....	14
Modify the appGetTree script to use Dictionary Values.....	15
additional code for the Folder name	15
Date Custom PProperty formatting Function	16
Implement the appgettree with folder script	16
Section 3: appGetTree Configuring Multiple Screens	17
appgettree with Multiple Screens	18
Launch the business application in chrome.....	18
Create the Application Plan	18
LEarnmode Application plan Designer.....	18
Configure and implement the appgettree script to create the debOut file	18
Configure appgettree to locate screen identifiers.....	19
Verify the Screen Identifiers	20
Configure the Screen identifiers	20



Section 1: **appGetTree**



APPGETTREE OVERVIEW

Perceptive Content 7.2.2.443 and 7.1.5.1899 introduced a new Learnmode method for learning values from a host application being accessed through a Chrome, Edge, or Firefox browser and copying those values to Perceptive Content document keys and custom properties. The new Learnmode method involves a VBScript function called appGetTree. All Perceptive Content versions after 7.2.2.443 support the use of the appGetTree VBScripted method of learnmode.

LAUNCH THE BUSINESS APPLICATION IN CHROME

1. Double-click the **PSI Web Demo** icon on the desktop.
2. Click the Select button next to Train User and then click the **Login as: Train User** link that was exposed.
3. Mouse over **Accounts Payable** and click **Invoice Information**.

NAVIGATE TO LEARNMODE APPLICATION PLANS

1. Launch the Perceptive Content think client and login with the following credentials:
 - a. Username: **inadmin**
 - b. Password: **train**
2. Verify the **Timed: 5 concurrent licenses with a 30 day runtime** radio button is selected and click OK.
3. Once the Perceptive Content Toolbar launches, click the manage icon.
4. In the Management Console that has launched, select **Accounts Payable** from the Select Department dropdown.
5. Expand the **Applications** option from the left-hand panel and select **LearnMode**.

(At this point you will need to size the Management Console and the Business Application launched in Chrome so you can see them side-by-side)

CREATE THE APPLICATION PLAN

1. Click the New... button.
2. Name your Application Plan **"PSIWebDemo"**
3. Click the Identify... button.
4. From the Window Selector Starter window click the **Start** button.
5. Move your cursor over the Business Application screen and single left-click.
6. Click the **OK** button.

LEARNMODE APPLICATION PLAN DESIGNER

1. Cancel or Close the Screen Properties window.
2. Select **Window Walker** from the Methods drop-down field.
3. Click **OK** to the Screen Properties window.
4. Click on the **Map** tab

CONFIGURE AND IMPLEMENT THE APPGETTREE SCRIPT TO CREATE THE DEBOUT FILE

1. Navigate to the **C:\Perceptive Content Installer Library\Learnmode Files\appGetTree** folder and open the **AppGetTree – Chrome 70.txt** in notepad++.
2. From Language menu option mouse over V and select visual basic
3. On line 12 replace the words Host Application with **Invoice Information**
4. Select all the script syntax and copy it (ctrl+A and cntrl+C)
5. Return to the Application Plan designer
6. Double-click the **Field1** document key
7. In the Filed1 Attributes window, change the Source field to **Literal** and type **Script Populated** in the Value field
8. Click the **Script:** field drop-down and select **Manage Scripts...**
9. Click the **Create** button in the Scripts window
10. Name your script **Invoice Information**
11. Select your script name and click the **Modify** button
12. Paste the copied script in the Script – Invoice Information window and click the **Test** button
13. Verify a **debout_0.txt** file is opened

LOCATING SCREEN DATA ON THE DEBOUT FILE AND SCRIPT CONFIG TO GRAB VALUE

The debut_X.txt file is used to find data that can be used as document keys or custom properties. It is the appGetTree equivalent to the Elements tab data in the designer when configuring a regular learnmode application plan. We will be learning the following values in the outlined Index structure:

Document Key or Custom Property	Value
Field1 (aka Vendor ID)	BOCA123
Field2 (aka Vendor Name)	Boise Cascade
Field3 (aka Invoice Number)	BC336
Custom Property: Invoice Amount	98.08
Custom Property: Invoice Date	07/04/2005

CONFIGURING APPGETTREE FOR FIELD1

1. With the **debout_0.txt** in focus, click **ctrl+F** and search for **BOCA123**

You will notice this value shows up in several strings as well as independently after the +tableCell or text[]-> values. While we can configure the script to locate any of these it is easiest to learn the data that shows up independently (e.g. +tableCell[BOCA123] or text[]->BOCA123 .

To learn a data value in the appGetTree script, we need to identify a unique row of syntax we will use as a point of reference (an anchor) above or below the data value we would like to learn. Once we locate this point of reference count the number of lines/rows above or below this point of reference that the data value you want to learn.

```
+tableCell[Vendor ID:]  
label[Vendor ID:]          This is our anchor row  
+tableCell[BOCA123] 1  
text[]->BOCA123 2      This is our data value row
```

For the Vendor ID we want to use as a document key we can count 2 lines/rows down from the anchor, **label[Vendor ID:]** to locate the value at the end of the text[]-> row. Remember this information

2. Maximize Notepad++ and add a single quote on the front of the syntax on line 21
3. Copy line 25 and paste it on line 26
4. Delete the single quote at the beginning of the copied line and **replace** the words **capturedValue** with **vendorID**
5. Within the syntax of line 26, **replace** the words **YOUR ANCHOR TAG HERE** with **label[Vendor ID:]**
(you can copy and paste the syntax from your debut_0.txt file)
6. **Delete** the **<+/- number of lines to desired value>** making sure not to delete the comma before it or the close parenthesis after. Type a **2** between the comma and close parenthesis
7. On line 27 type the following: **Field1=vendorID**

```
26 vendorID = GetValueFromAnchor(treeInfo(1), "label[Vendor ID:]", 2)  
27 Field1=VendorID
```


8. Select all of the updated script and copy it
9. Open the Application Plan designer and in the Script – Invoice Information window, select all of the script and paste in the copy
10. Click the **Test** button and verify that the test field is populated with **BOCA123**

CONFIGURING APPGETTREE FOR FIELD2

1. With the **debout_0.txt** in focus, click **ctrl+F** and search for **Boise Cascade**

You will notice this value shows up in several strings as well as independently after the +tableCell or text[]-> values. While we can configure the script to locate any of these it is easiest to learn the data that shows up independently (e.g. +tableCell[Boise Cascade] or text[]->Boise Cascade .

To learn a data value in the appGetTree script, we need to identify a unique row of syntax we will use as a point of reference (an anchor) above or below the data value we would like to learn. Once we locate this point of reference count the number of lines/rows above or below this point of reference that the data value you want to learn.

```
+tableCell[Vendor Name:]
  label[Vendor Name:]
+tableCell[Boise Cascade]
  text[]->Boise Cascade
```

This is our anchor row

1

2

This is our data value row

For the Vendor Name we want to use as a document key we can count 2 lines/rows down from the anchor, **label[Vendor Name:]** to locate the value at the end of the text[]-> row. Remember this information

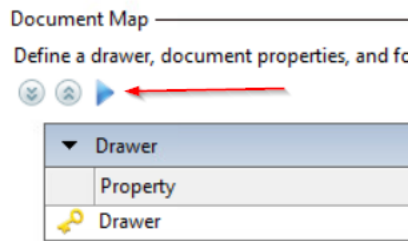
2. Maximize Notepad++ and add a single quote on the front of the syntax on line 21
3. Copy line 25 and paste it on line 29
4. Delete the single quote at the beginning of the copied line and **replace** the words **capturedValue** with **vendorName**
5. Within the syntax of line 29, **replace** the words **YOUR ANCHOR TAG HERE** with **label[Vendor Name:]** (you can copy and paste the syntax from your debut_0.txt file)
6. **Delete** the **<+/- number of lines to desired value>** making sure not to delete the comma before it or the close parenthesis after. Type a **2** between the comma and close parenthesis
7. On line 30 type the following: Field2=vendorName

```
29 vendorName= GetValueFromAnchor(treeInfo(1), "label[Vendor Name:]", 2)
30 Field2=vendorName
```

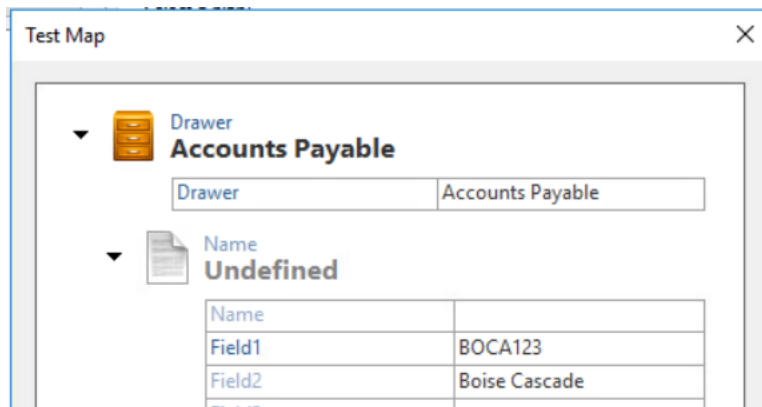
8. Select all of the updated script and copy it
9. Open the Application Plan designer and in the Script – Invoice Information window, select all of the script and paste in the copy
10. Click the **Test** button and verify that the test field is populated with **BOCA123**

Because the appGetTree script is being used on field1 we will have to test field2 differently

11. Click the **OK** button on the Script – Invoice Information window and the Scripts window
12. In the Field1 Attributes window, **select** the **Invoice Information** script from the Script: drop-down field
13. Click the **OK** button
14. Click the play button (blue triangle) below the Document Map header



15. Verify your results:



16. Click the **Close** button on the Test Map window
17. Double-click the **Field1** under the Property column in the Map section of the Application Plan Designer
18. Click the **ellipsis** button next to the Script: field drop-down. This will bring the Script – Invoice Information window back up for additional updates

CONFIGURING THE APPGETTREE SCRIPT FOR THE REMAINING FIELDS

1. Using the knowledge gained from the previous steps configure the script to learn the remaining index keys and custom properties.

Index Key	Value to Learn	Script Syntax
Field3	BC336	Field3=invNumber
CP: Invoice Amount	98.08	DocProperty("Invoice Amount")=invAmount
CP: Invoice Date	07/04/2005	DocProperty("Invoice Date")=invDate

2. Test your configuration by using steps 13 through 18 from the previous exercise
(Be aware that the Invoice Amount custom property value will not show up during testing. To fully test everything we will link a document with this application plan.)

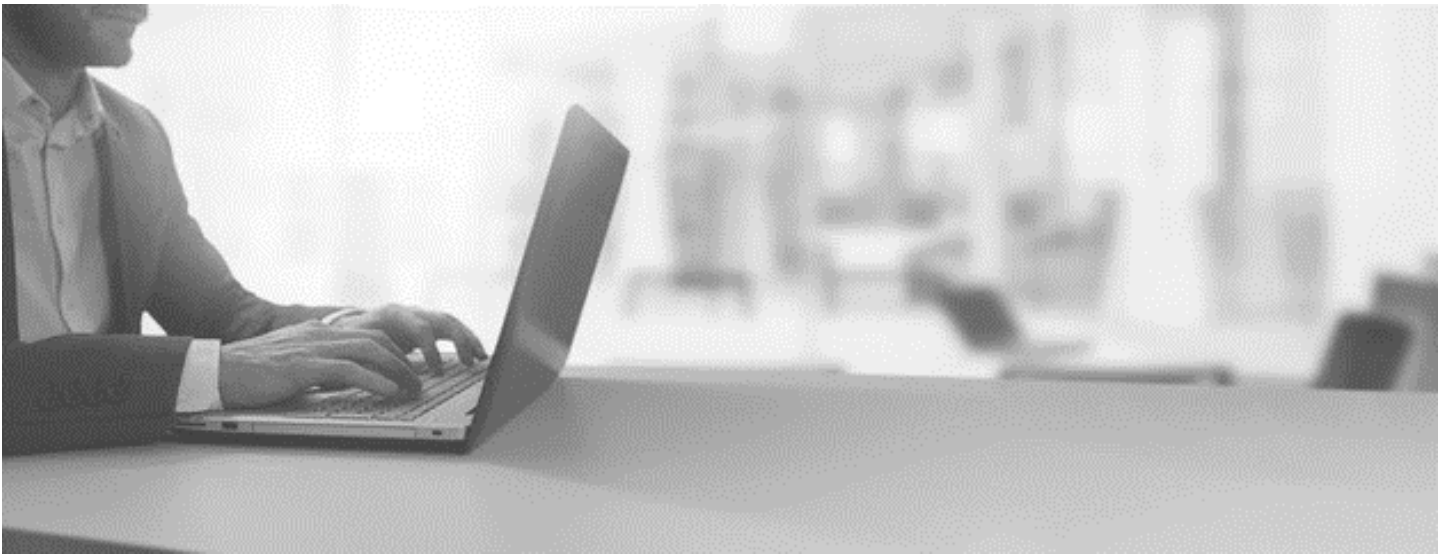
UPDATE APPLICATION PLAN DOCUMENT KEY LABELS

While testing your application plan you may have noticed that the Document Key Fields are using the generic field labels (Field1, Field2...) instead of a field label that reflects the data in each of the specific fields. We will configure alternate labels in the proceeding steps.

1. Double-click **Field1** in the Application Plan Designer Map section
2. In the Field1 Attributes window, update the **Alternate Label:** field with **Vendor ID**
3. Click the OK button
4. Double-click **Field2** in the Application Plan Designer Map section
5. Change the **Source** field to **Literal** and in the exposed **Value:** field type **Script Populated**
6. Update the **Alternate Label:** field with **Vendor Name**
7. Double-click **Field3** in the Application Plan Designer Map section
8. Change the **Source** field to **Literal** and in the exposed **Value:** field type **Script Populated**
9. Update the **Alternate Label:** field with **Invoice Number**
10. Test by clicking the play button (blue triangle) below the Document Map header section

TEST THE APPLICATION PLAN

1. **Save** your application plan from the application plan designer
2. Verify the PSI Web Demo application (Business Application) is on an **Accounts Payable > Invoice Information** Screen
3. Log out of the Perceptive Content client and log back in.
Username: **inadmin**
Password: **train**
4. Open the Demo Images folder on your desktop
5. **Drag and drop** one of the **Demo XX.tif** images onto the Perceptive Content toolbar
6. In the New Document.... window, under the Select an application plan field, select the **PSIWebDemo** application plan.
7. Verify all Document Key and Custom Property fields are being populated and the Document Key Labels are correct.



Section 2: **appGetTree with Folders**



appGetTree Configuration for Folders

APPGETTREE WITH FOLDERS OVERVIEW

In this section will copy and reconfigure our current appGetTree script to be used on an application plan configured to automatically create a configured shortcut map during linking and automatically add the document shortcut to the defined folder.

LAUNCH THE BUSINESS APPLICATION IN CHROME

1. Double-click the **PSI Web Demo** icon on the desktop.
2. Click the Select button next to Train User and then click the **Login as: Train User** link that was exposed.
3. Mouse over **Accounts Payable** and click **Invoice Information**.


CREATE THE APPLICATION PLAN

1. From the Perceptive Content Management Console, **Application Plans > LearnMode**, Click the **New...** button.
2. Name your Application Plan **PSIWebDemoFolders**
3. Click the **Identify...** button.
4. From the Window Selector Starter window click the **Start** button.
5. Move your cursor over the Business Application screen and single left-click.
6. Click the **OK** button.

LEARNMODE APPLICATION PLAN DESIGNER

1. **Cancel or Close** the **Screen Properties** window.
2. Select **Window Walker** from the Methods drop-down field.
3. Click **OK** to the Screen Properties window.
4. Click on the **Map** tab

CREATE DICTIONARY VALUES

1. From the Dictionary panel toolbar, click the **New Field** icon 
2. In the Dictionary window **Name:** field type **Vendor ID**
3. Click the **OK** button

4. Repeat steps 1-3 to create the following values:

- a. **Vendor Name**
- b. **Invoice Number**
- c. **Invoice Amount**
- d. **Invoice Date**
- e. **Folder Name**

Dictionary	
Name	Example Data
Vendor ID	
Vendor Name	
Invoice Number	
Invoice Amount	
Invoice Date	
Folder Name	

MAPPING DICTIONARY VALUES

- From the **Dictionary** panel, left-click hold and **drag and drop** the **Vendor ID** Dictionary value onto the **Field1** row within the Map screen.
- Repeat step 1 to Map the following Dictionary values:

Dictionary Value	Mapped To
Vendor Name	Field2
Invoice Number	Field3
Invoice Amount	Invoice Amount (Custom Property)
Invoice Date	Invoice Date (Custom Property)

The screenshot shows the Dictionary panel on the left and the Map section on the right. Red arrows indicate the following mappings:

- Vendor ID (Field1) from the Dictionary panel to the Map section.
- Vendor Name (Field2) from the Dictionary panel to the Map section.
- Invoice Number (Field3) from the Dictionary panel to the Map section.
- Invoice Amount from the Dictionary panel to the Map section.
- Invoice Date from the Dictionary panel to the Map section.


The Map section shows a list of properties and their sources:

Property	Source
Document Name	Undefined
Vendor ID (Field1)	Dictionary
Vendor Name (Field2)	Dictionary
Invoice Number (Field3)	Dictionary
Field4	Undefined
Field5	Undefined
Document Type	Document Type
Invoice Amount	Dictionary
Invoice Date	Dictionary

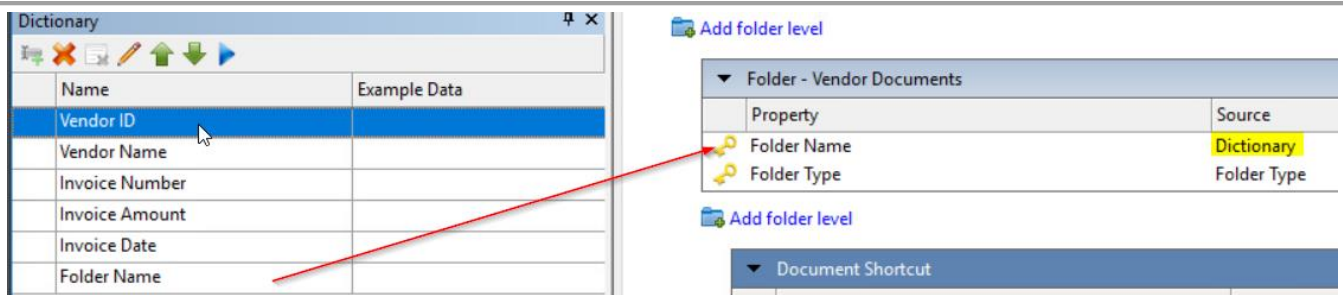
Shortcut Map —
Define a drawer and folder levels to create a shortcut for the item in the last level of the map

CREATE A SHORTCUT MAP


- From the Map section of the Application Plan Designer, click **Add shortcut map**

Shortcut Map —
Define a drawer and folder levels to
 **Add shortcut map**

- From the Dictionary panel, left-click **drag and drop** the **Folder Name** value onto the **Folder Name** in the Shortcut Map



MODIFY THE APPGETTREE SCRIPT TO USE DICTIONARY VALUES

1. Pull up the previously configured appGetTree script in Notepad++
2. Select all of the script syntax (**Ctrl+A**) and copy it (**Ctrl+C**)
3. Create a new tab in Notepad++ by clicking the **New** button from the  toolbar
4. **Paste** in the copied script into the editing section of the newly created tab
5. On line 27, **replace** the **Field1** syntax with **Field("Vendor ID")**

```
Field("Vendor ID")=vendorID
```

The Vendor ID syntax between the double quotes within the Field("XXXXXXX") variable is referencing the Dictionary value we created in the previous steps.

6. Update the remaining lines of the appGetTree script using the proceeding screenshot:

```
vendorID= GetValueFromAnchor(treeInfo(1), "label[Vendor ID:]", 2)
Field("Vendor ID")=vendorID

vendorName= GetValueFromAnchor(treeInfo(1), "label[Vendor Name:]", 2)
Field("Vendor Name")=vendorName

invNumber= GetValueFromAnchor(treeInfo(1), "label[Invoice Number:]", 2)
Field("Invoice Number")=invNumber

invAmount= GetValueFromAnchor(treeInfo(1), "label[Total Amount:]", 2)
Field("Invoice Amount")=invAmount

invDate= GetValueFromAnchor(treeInfo(1), "label[Invoice Date:]", 2)
Field("Vendor Date")=invDate
```

ADDITIONAL CODE FOR THE FOLDER NAME

1. In a row below the row with Field("Invoice Date") variable, add the following syntax:

```
Field("Folder Name")=vendorID+ " - "+vendorName
```

```
invDate= GetValueFromAnchor(treeInfo(1), "label[Invoice Date:]", 2)
Field("Invoice Date")=invDate

Field("Folder Name")=vendorID+ " - "+vendorName
```

Since we are going to use a concatenation of the Vendor ID and Vendor Name values to dynamically name our folders, we can reuse the variables from the proceeding code to grab those values (vendorID and vendorName).

DATE CUSTOM PROPERTY FORMATTING FUNCTION

Because the Invoice Date is a Date Type Custom Property, we will need to add a function to define the current format of the date so it can be converted to a format that can be stored in a Date Type data table column in the INOW database. This function is called DoMacroDateConversion (eg vardate= DoMacroDateConversion(varInitialDate,"MM/dd/yyyy")).

1. Add a row **between** the rows that begin with invDate... and **Field("Invoice Date")...**

2. **Type** the following code:

```
invDateNew= DoMacroDateConversion(invDate,"MM/dd/yyyy")
```

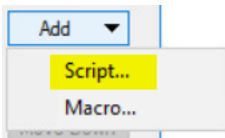
3. **Update** the **invDate** variable on the next line to **invDateNew**

```
invDate= GetValueFromAnchor(treeInfo(1), "label[Invoice Date:]", 2)
invDateNew=DoMacroDateConversion(invDate,"MM/dd/yyyy")
Field("Invoice Date")=invDateNew
```


IMPLEMENT THE APPGETTREE WITH FOLDER SCRIPT

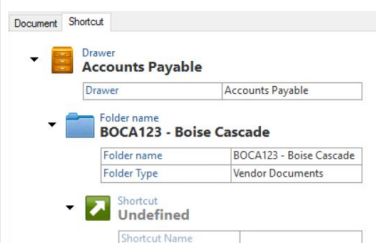
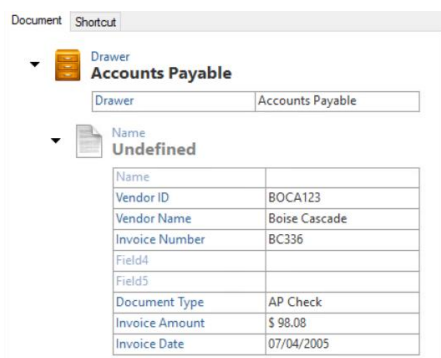
Unlike our previous appGetTree script we placed on the Field1 value within the Application Plan Designer, this script will be placed on a single Dictionary value.

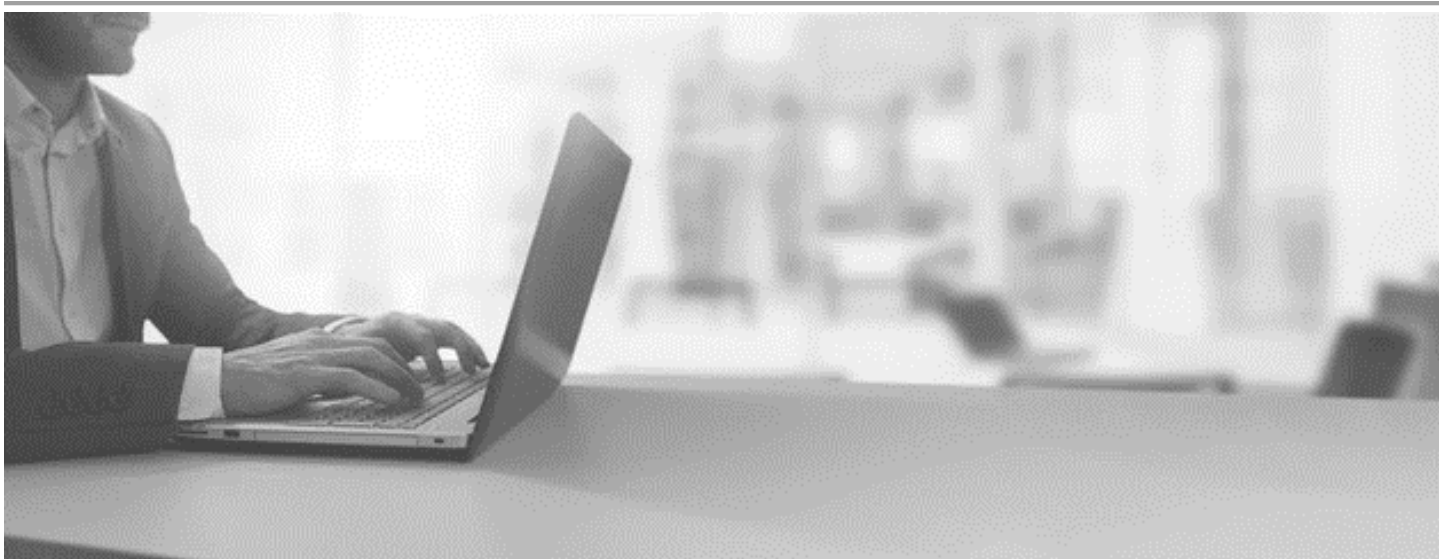
1. Select all and **copy** your **appGetTree** script code from Notepad++
2. Go to or open the **PSIWebDemoFolder application plan** in the Application Plan Designer
3. **Double-click** the **Vendor ID** in the Dictionary panel
4. In the Dictionary Field window, click the **Add** drop-down button and select **Script**



5. In the Scripts window click the **Create** button
6. Name the script **Invoice Information with Folders**
7. **Select** the **Invoice Information with Folders** script from the list and click the **Modify** button
8. In the Script – Invoice Information with Folders window Script field, **paste** the copied **appGetTree** script
9. Click **OK** in each of the open windows.

10. **Test** the script by going to the Map section in the Application Plan Designer and **clicking** the **play** button  below the Document Map and Shortcut Map header sections. Results should be the same as the following screenshots





Section 3: **appGetTree** Configuring Multiple Screens

APPGETTREE WITH MULTIPLE SCREENS

Analogous to non-scripted learnmode application plans, appGetTree also allows configuration for multiple business application screens. A typical use case for multiple screens is configuring one screen for linking and viewing (via the use of the binoculars button) and additional screens users can also use the binoculars button on the Perceptive Content toolbar. These additional screen configuration is commonly referred to as view enabling a screen. In this lab we will be learning the Accounts Payable > Invoice Information and Time and Expenses > Expense Entry screen.

LAUNCH THE BUSINESS APPLICATION IN CHROME

2. Double-click the **PSI Web Demo** icon on the desktop.
3. Click the Select button next to Train User and then click the **Login as: Train User** link that was exposed.
4. Mouse over **Accounts Payable** and click **Invoice Information**.

CREATE THE APPLICATION PLAN

7. From the Perceptive Content Management Console, **Application Plans > LearnMode**, Click the **New...** button.
8. Name your Application Plan **PSIWebDemoMultiScreens**
9. Click the **Identify...** button.
10. From the Window Selector Starter window click the **Start** button.
11. Move your cursor over the Business Application screen and single left-click.
12. In the Plan Setting window, update the **Value:** field to **::*** (get rid of everything after the beginning double colons and add a asterisk)
13. Click the **OK** button.

LEARNMODE APPLICATION PLAN DESIGNER

5. **Cancel or Close** the **Screen Properties** window.
6. Select **Window Walker** from the Methods drop-down field.
7. Click **OK** to the Screen Properties window.
8. Click on the **Map** tab

CONFIGURE AND IMPLEMENT THE APPGETTREE SCRIPT TO CREATE THE DEBOUT FILE

1. From a windows folder explorer go to the **C:\Perceptive Content Installer Library\Learnmode Files\appGetTree** folder and open the **appGetTree multiple-screens.txt** in Notepad++.
2. From **Language** menu option mouse over **V** and select **visual basic**
3. On line 18 replace the words **Host Application** with **::**
4. Select all the script syntax and copy it (ctrl+A and cntrl+C)
5. Return to the Application Plan designer
6. Double-click the **Field1** document key
7. In the Filed1 Attributes window, change the Source field to **Literal** and type **Script Populated** in the Value field
8. Click the **Script:** field drop-down and select **Manage Scripts...**

9. Click the **Create** button in the Scripts window
10. Name your script **MultiScreen**
11. Select your script name and click the **Modify** button
12. Paste the copied script in the Script – MultiScreen Information window and click the **Test** button
13. Verify a **debout_X.txt** file is opened
14. Change the **PSI Web Demo** to the **Time & Expense > Expense Entry** screen
15. **Click** the **Test** button again and verify a second **debout_X.txt** file is opened.

Keep these debout files opened as they can be used to help find screen identifiers (referred to as the linking screen in the appGetTree script) as well as document keys and custom properties while configuring the appGetTree script.

16. **Add a single quote** to the beginning of **line 24** to disable the script from creating additional debout files

CONFIGURE APPGETTREE TO LOCATE SCREEN IDENTIFIERS

Use the debout files to locate a value that uniquely identifies the screen the debout file was created from when you ran the script.

1. Go to **line 67** in the appGetTree script and **remove** the **single quote** at the beginning of the line
2. **Replace** the syntax **YOUR ANCHOR TAG HERE** with **+menuItem[System]**
3. **Delete** the **<+/- number of lines to desired value for screen identifier>** making sure not to delete the comma before it or the close parenthesis after. Type a **1** between the comma and close parenthesis

```
linkingScreen = GetValueFromAnchor(treeInfo(1), "+menuItem[System]", 1)
```

Because this configuration grabbing a long string values to use as the identifier of the window, we will use a Split function to only use a portion of the string as the identifier (e.g. Accounts Payable versus :: Accounts Payable :: Invoice Information – Google Chrome)

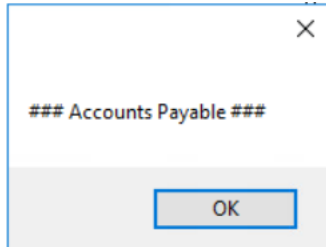
```
+menuBar[System]
+menuItem[System]
+titleBar[]->:: Accounts Payable :: Invoice Information - Google Chrome
titleBar[]->:: Accounts Payable :: Invoice Information - Google Chrome
titleBar[]->:: Accounts Payable :: Invoice Information - Google Chrome
titleBar[]->:: Accounts Payable :: Invoice Information - Google Chrome
titleBar[]->:: Accounts Payable :: Invoice Information - Google Chrome
+pane[:: Accounts Payable :: Invoice Information - Google Chrome]
```

4. Go to **line 75** and **add** the following function configuration after the equals sign: **Split(linkingScreen,"::")(1)**
- ```
GetLinkingScreen = Split(linkingScreen,"::") (1)
```
5. Go to **line 20** and **remove** the **single quote** at the beginning of the line
  6. **Select all** and **copy** the entire script in Notepad++
  7. In the **Script – MultiScreen** window, **replace** the **old script** with the newly copied script

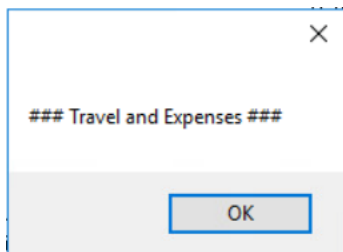
---

## VERIFY THE SCREEN IDENTIFIERS

1. Make sure you are on the **PSI Web Demo Accounts Payable > Invoice Information** application screen
2. Click the **Test** button from the **Script – MultiScreen window**
3. Verify a message box that says **### Accounts Payable ###** pops up, then click OK



4. Go to the **PSI Web Demo Travel & Expense > Expense Entry** application screen
5. Click the **Test** button from the **Script – MultiScreen window**
6. Verify a message box that says **### Travel and Expenses ###** pops up, then click OK



Please take note of the screen identifiers returned in the message boxes

## CONFIGURE THE SCREEN IDENTIFIERS

1. Go to **line 20** and **add a single quote** at the beginning of the line
2. Go to **line 25** and **remove the single quote**  
Line 25 uses a Trim function to remove any preceding and proceeding spaces around the screen identifiers
3. Go to **line 27** and **replace** the syntax **SCREEN IDENTIFIER 1 HERE** with **Accounts Payable**  

```
IF varScreen = "Accounts Payable" THEN
```
4. Go to **line 41** and **replace** the syntax **SCREEN IDENTIFIER 2 HERE** with **Travel and Expenses**  

```
ELSE IF varScreen = "Travel and Expenses" THEN
```
5. Configure the script to get values you want to use as document keys below the respective IF Then screen identification lines. See proceeding screenshots:

```

IF varScreen = "Accounts Payable" THEN

 ' this uses "YOUR ANCHOR TAG HERE" as the anchor and indexes the capturedValue off this anchor tag.
 'capturedValue = GetValueFromAnchor(treeInfo(1), "YOUR ANCHOR TAG HERE", <+/- number of lines to desired value>)
 vendorID= GetValueFromAnchor(treeInfo(1), "label[Vendor ID:]", 2)
 Field1=vendorID

 vendorName= GetValueFromAnchor(treeInfo(1), "label[Vendor Name:]", 2)
 Field2=vendorName

 invNumber= GetValueFromAnchor(treeInfo(1), "label[Invoice Number:]", 2)
 Field3=invNumber

 invAmount= GetValueFromAnchor(treeInfo(1), "label[Total Amount:]", 2)
 DocProperty("Invoice Amount")=invAmount

 invDate= GetValueFromAnchor(treeInfo(1), "label[Invoice Date:]", 2)
 DocProperty("Invoice Date")=invDate

ELSE IF varScreen = "Travel and Expenses" THEN

 ' this uses "YOUR ANCHOR TAG HERE" as the anchor and indexes the capturedValue off this anchor tag.
 'capturedValue = GetValueFromAnchor(treeInfo(1), "YOUR ANCHOR TAG HERE", <+/- number of lines to desired value>)
 empExpense = GetValueFromAnchor(treeInfo(1), "label[Total Employee Expenses:]", 2)
 Field1=empExpense

```

You can copy sample document key and custom property configuration for each screen from the **C:/Perceptive Content Installer Library/Learnmode Files/appGetTree/DocKeyConfigurationSamples.txt**

6. Once the document keys and custom properties well configured, **select all** and **copy** your script.
7. In the **Script – MultiScreen** window **replace** the **existing script** with the newly copied script
8. **Test** by clicking the blue play buttons in the Map section of the application plan designer. Make sure you **test** with both the **Accounts Payable > Invoice Information** and **Time & Expense > Expense Entry** screens in the **PSI Web Demo** business application