Oracle® Distributed Document Capture

Developer's Guide

Release 10*g*R3

E13868-01

November 2010

Developer's Guide for Oracle Distributed Document Capture, Release 10*g*R3

E13868-01

Primary Author: Sarah Howland

Contributor: Ken Peterka, Rob Abbe, Dan Sievers, Jun Liang, Sara Johnson, Carl Diedrich, Vince Cook, and Richard Lindman

Contents

Preface

The *Developer's Guide for Oracle Distributed Document Capture* contains information to develop VBScript macros to customize the Oracle Distributed Document Capture application for your organization. Oracle Distributed Document Capture uses the VBScript engine built into Microsoft's Internet Explorer.

Audience

This document is intended for developers responsible for customizing Oracle Distributed Document Capture functionality.

Related Documents

For more information, see the following documents in the Oracle Distributed Document Capture Release 10*g*R3 documentation set:

- *Oracle Distributed Document Capture Release Notes*
- *Installation Guide for Oracle Distributed Document Capture*
- *Administrator's Guide for Oracle Distributed Document Capture*
- *User's Guide for Oracle Distributed Document Capture*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Using Scripts in Oracle Distributed Document Capture

This section covers the following topics:

- "About Oracle Distributed Document Capture Scripts" on page 1-1
- "Creating Client Scripts" on page 1-1
- "Integrating the Client With Other Web Applications" on page 1-3
- "Debugging Client Scripts" on page 1-5

1.1 About Oracle Distributed Document Capture Scripts

Oracle Distributed Document Capture scripts enable you to provide extended functionality to client users. For example, use scripts to:

- Perform custom data integrity checking while indexing.
- Dynamically populate a field pick-list while indexing.
- Automatically populate document index fields before sending a batch.
- Prevent the sending of a batch based on custom criteria.

Oracle Distributed Document Capture uses the VBScript engine built into Microsoft's Internet Explorer. This design helps to streamline the client by maintaining a lighter weight client interface. Administrators can assign a VBScript file to one or many scan profiles. Because scripts are managed centrally, they are automatically downloaded to clients during login. Once downloaded, they run within Internet Explorer.

1.2 Creating Client Scripts

This section covers the following topics:

- "Using the Template VBScript File" on page 1-1
- "Copying VBScripts to the Scripts Folder" on page 1-2
- "Creating and Assigning Client Scripts" on page 1-2
- "Linking Multiple VBScripts" on page 1-3

1.2.1 Using the Template VBScript File

After installation, a template VBScript file called *Template.VBS* is copied into the WebPages folder. It is recommended that you use this file as a base for new scripts. The Template.VBS file contains procedure stubs for all events called by the client.

These procedure stubs must be present in your VBScripts. If your script is missing an event procedure stub, a runtime error will occur. In addition to the client event procedures, you can develop as many private procedures as you like.

1.2.2 Copying VBScripts to the Scripts Folder

After installing Oracle Distributed Document Capture, the physical folder name for the Capture virtual directory is WebPages (for example, c:\Program Files\Oracle\Document Capture\WebPages). In order to assign a VBScript to a scan profile, the script must reside in a folder called *Scripts* within the Capture virtual directory:

- Physical path:

  ```
  c:\Program Files\Oracle\Document Capture\WebPages\Scripts
  ```

- Web path:

  ```
  www.domainname.com\Capture\Scripts
  ```

> **Note:** Your system administrator may have changed the location of the Capture virtual directory.

1.2.3 Creating and Assigning Client Scripts

Follow these steps to create a client script and assign it to a scan profile:

1. From the WebPages folder, make a copy of the Template.VBS script file.

   For example, name the new version *WebCaptureScript1.VBS*.

2. Place the new script file into the Scripts folder.

   (See "Copying VBScripts to the Scripts Folder" on page 1-2.)

3. Open the file using Windows Notepad or another text editor.

4. Write your custom script within the Event Procedure stubs.

   For example, write code in the first called event named *ecNet_ScriptStart*.

   > **Note:** Objects and client events relating to Oracle Distributed Document Capture are referred to as Web Capture objects and Web Capture events. They are listed in Chapter 2, "Web Capture Objects and Client Events".

5. Save the script file.

6. Assign the script file to a scan profile.

   Open a scan profile. On the General node, select the script file in the **Profile Script** field.

7. Log into the client and select the scan profile that will run your script.

   If a runtime error occurs, see "Debugging Client Scripts" on page 1-5.

1.2.4 Linking Multiple VBScripts

By default, a VBScript cannot include references to other VBScript modules (i.e., *.VBS files). To help promote code reuse, the server includes a VBScript compiler that allows you to include procedures from different VBScript files.

To use this feature, use the Include directive. The following is an example:

```
'#INCLUDE:"c:\program files\oracle\document capture\webpages\util.vbs"
'#INCLUDE:"c:\program files\oracle\document capture\webpages\mycode.vbs"
```

Rules for the Include directive:

1. The statement must appear at the top of a script file.
2. The apostrophe in front of the directive is required.
3. You can have any number of Include directives, but they must be grouped together at the top of a script file.
4. Only one file may be referenced per Include directive.
5. The IIS process must have read access to the folder in which the macro resides. (This is the default when the virtual directory is set up.)
6. It is possible to nest Include directives. In the example above, the Util.VBS file may also have Include directives.

In VBScript, the apostrophe indicates a comment line. Since the Include directive is a feature of Oracle Distributed Document Capture, it cannot reside on a non-commented line. Therefore, the apostrophe is required.

If you use the Include directive, be especially careful to ensure that procedure names and variable declarations do not conflict among the included files. When a client accesses a scan profile that uses a VBScript that contains the Include Directives, the server compiles the VBScript by combining all Included files into a single (i.e., larger) VBScript file. This larger VBScript file is then downloaded to the client. The server is optimized for performance to ensure that the compilation process only occurs if the VBScript files have been updated.

1.3 Integrating the Client With Other Web Applications

You can run the client from another web application. The client is invoked via the web address and parameters such as the scanning profile and optional index values are passed within the address.

For example, you might add a Scan button to a line of business web application. After completing entry fields, the user clicks Scan. The client immediately begins scanning a document, using settings in the scan profile specified in the web address. Once scanned, the document is displayed in the Review/Index screen. Index fields are automatically populated with user entries, which were passed in the web address. The user reviews the document and completes other index fields, then either sends the batch or closes it. Upon close, the user might return to the line of business web application.

Important points about the client integration:

- Client users must be authenticated using Web Server Authenticated Access rather than via Oracle Distributed Document Capture Prompted Login. This allows the client to launch automatically without users needing to log in.

- You must specify a scan profile with a type of 2 - Scan and Index Documents. This creates a single document in the batch, although users can create or remove documents in the Review/Index screen.

Use the properties listed in Table 1–1 to configure a client integration.

Table 1–1 Client Integration ActiveX Properties

Property	Description
Profile	Specifies the scan profile to be used. The scan profile must be a 2- Scan and Index Documents type profile.
ScanAction	Specifies the action to be taken, where: ■ 0 - None (default): No action is taken. ■ 1 - Scan: Scanning begins into a new batch using the scan profile specified in the Profile property. Once complete, the batch is displayed in the Review/Index screen to the user. ■ 2 - Import: Importing begins into a new batch using the scan profile specified by the Profile property. Once complete, the batch is displayed in the Review/Index screen to the user.
IndexData	Specifies index field name/value pairs delimited by tab characters. If specified, values in this property are applied to the document.
URLOnclose	If specified with a web address as its value, this redirects the browser window to the specified value. This method is useful when the webcapture.asp file is embedded in another page using IFrame. If specified with the value *CloseBrowserOnExit*, the browser window will be closed. This method is useful when the webcapture.asp is launched in a new window.
CloseBrowserOnSendPrompt, CloseBrowserOnSendNoPrompt	If either property is specified, the browser closes after sending a document or if the user cancels or closes the browser at any point. If you specify *CloseBrowserOnSendPrompt* and a batch was created, the user is prompted to save the batch before the browser is closed.

Example Client Integration Web Address

Here is an example webcapture.asp URL. (Note that this URL should be all on one line.)

```
http://localhost/capture/WebCapture.Asp?
Profile=Contracts&ScanAction=1&IndexData=Company%09Acme%09DocTyp
e%09Contract&URLOnclose=CloseBrowserOnExit
```

This web address configures the client integration as follows:

- Profile=Contracts: The Contracts scan profile will be used for scanning and indexing.
- ScanAction=1: The control will attempt to immediately capture a document from the scanner's ADF.
- IndexData: The Company index field will be populated with the value *Acme* and the DocType field will be populated with the value *Contract*. The %09 characters represent a Tab character.

- URLOnclose=CloseBrowserOnExit: After the user closes the Client screen, the browser window will be closed.

Unrecognized Parameters Treated as Index Fields

Unrecognized parameters found on the URL are treated as index fields. For example, in the URL that follows, the *Last Name* parameter is treated as an index field and is added to the data passed in through the IndexData parameter. (This accommodates applications such as Oracle Imaging and Process Management (I/PM) that generate URLs that cannot contain the tab characters required by the IndexData parameter.)

```
http://localhost/capture/WebCapture.Asp?Profile=Contracts&ScanAc
tion=1&IndexData=Company%09Acme%09DocType%09Contract&URLOnclose=
CloseBrowserOnExit&Last%20Name=Doe
```

1.4 Debugging Client Scripts

If Microsoft's Visual Studio is installed on the client workstation, it can be used in debugging VBScript. When a runtime error occurs in a script and Visual Studio is installed, a Microsoft Development Environment message displays, asking if you would like to debug the application.

Click Yes to open the script in Visual Studio. When prompted to create a new project, click No.

Visual Studio displays the VBScript source code in the Microsoft Development Environment.

Use the Visual Studio debugger to trace the execution of the script and inspect the value of variables. Although Visual Studio does not allow you to modify the script, it is an invaluable tool in debugging your VBScripts. Once you determine how to correct a runtime or logical program error, modify the script on the server using Notepad or other text based editor. After the VBScript has been updated on the server, log back into the client to download the updated VBScript.

2 Web Capture Objects and Client Events

This section covers the following topics:

- "Web Capture Objects" on page 2-1
- "Web Capture Client Events" on page 2-13

2.1 Web Capture Objects

Web Capture objects allow Oracle Distributed Document Capture macros to obtain batch and scan profile information.

2.1.1 ecnBatch Object

Description: This object represents a batch in Oracle Distributed Document Capture.

Properties	Type	Description
BatchFilePath	String	Returns the path of the batch images.
BatchName	String	Returns the name of the batch (less the prefix).
BatchPath	String	Sets/returns the subdirectory of the batch. BatchPath is the parent directory of BatchFilePath.
BatchPrefix	String	Returns the prefix portion of the batch name.
CommitMethod	Integer	0 to not commit, 1 to use the commit server, 2 to commit immediately.
ComputerName	String	Returns the computer name that created the batch.
CreatedDate	String	Returns the date on which the batch was created.
CreateUser	String	Returns the name of the user who created the batch.
CurrentPriority	Integer	Returns the priority of the batch.
CurrentStatus	String	Returns the status of the batch as set up in the scan profiles for the batch.
Dirty	Boolean	Returns True if the batch needs to be saved.
Documents	Integer	Collection of document objects for the batch.
FileCabinet	String	Returns the name of the batch's file cabinet.
Indexed	Boolean	Returns True if the batch is considered indexed.

Properties	Type	Description
LastImgNum	Integer	Sets/returns the image naming scheme for the last image in the batch. For example, a two page batch will contain IMG001.tif, IMG002.tif. 2 will be the return value of LastImgNum property. This is used internally in the program for batch page naming purposes.
ModifiedDate	String	Returns the date on which the batch was last modified.
ModifyUser	String	Returns the name of the user who last modified the batch.
Note	String	Returns the note text for the batch.
Profilename	String	Returns the name of the profile used to create the batch.
ScanningType	Long	Sets/returns Scanning Type specified for the profile that created the batch. 0 - Scan Only 1 - Scan and Index Batch 2 - Scan and Index Document

2.1.2 ecnBatches Object

Description: This object represents a collection of batches.

Properties	Type	Description
Item (Index As Variant)	ecnBatch	Returns an ecnBatch reference based on the Index specified.
Count	Long	Returns the number of ecnBatches in the collection.

Method	Add
Description	Adds an ecnBatch reference into the collection.
Syntax	Add (ByVal Key As String, Optional sKey As String) As ecnBatch
Parameters	AKey - Value will be assigned to ecnBatch.Key property, and will be used as a Key to identify the item within the collections.

Method	Clear
Description	Clears out all batches within the collection.
Syntax	Clear ()
Parameters	None

Method	Remove
Description	Removes a specific ecnBatch out of the collection.
Syntax	Remove (Index As Variant)
Parameters	Index - Key to identify the ecnBatch object within the collection

2.1.3 ecnDocument Object

Description: This object represents a document in Oracle Distributed Document Capture.

Properties	Type	Description
Fields	Object	Returns a reference to the fields collection consisting of ecnField objects.
Images	Object	Returns a reference to the images collection for the batch consisting of ecnImage objects.
Key	String	Returns the document key.

2.1.4 ecnDocuments Object

Description: This object represents a collection of documents within a batch.

Properties	Type	Description
Item (Index As Variant)	ecnDocument	Returns an ecnDocument reference based on the Index specified.
Count	Long	Returns the number of ecnDocuments in the collection.

Method	Add
Description	Adds an ecnDocument reference into the collection.
Syntax	Add (Key As String) As ecnDocument
Parameters	Key - Not Used

Method	Add2
Description	Adds an ecnDocument reference into the collection at a specific location.
Syntax	Add2 (Key As String, oBefore As ecnDocument, oAfter As ecnDocumennt) As ecnDocument
Parameters	Key - Not Used. oBefore - a document reference where the new document should be added one spot before it in the collection. oAfter - a document reference where the new document should be added one spot after it in the collection.

Method	Clear
Description	Clears out all documents within the collection.
Syntax	Clear ()
Parameters	None

Method	Remove
Description	Removes a specific document out of the collection.
Syntax	Remove (Index As Variant)
Parameters	Index - Key to identify the document object within the collection

2.1.5 ecnField Object

Description: This object represents a document field.

Properties	Type	Description
FieldName	String	Returns the name of the index field.
Key	String	Returns the field Key.
Length	Integer	Returns the maximum length of the field.
Required	Boolean	Returns True if the field must have a value before commit.
Value	String	Returns the present value of the field.

2.1.6 ecnFields Object

Description: This object represents a collection of fields within a document.

Properties	Type	Description
Item (Index As Variant)	ecnField	Returns an ecnField reference based on the Index specified.
Count	Long	Returns the number of ecnFields in the collection.

Method	Add
Description	Adds an ecnField reference into the collection.
Syntax	Add (ByVal Key As String, Optional sKey As String) As ecnField
Parameters	Key - value will be assigned to ecnField.Key property, and will be used as a Key to identify the item within the collections. sKey - Not Used

Method	AddField
Description	Adds a field to the collection of fields.
Syntax	AddField(oField As ecnField, Optional sKey As String)
Parameters	ecnField is an ecnField object. (See "ecnField Object" on page 2-4.) sKey is the value that will be assigned to ecnField.Key property, and will be used as a key to identify the item within the collections.

Method	Clear
Description	Clears out all fields within the collection.
Syntax	Clear ()
Parameters	None

Method	Remove
Description	Removes a specific field out of the collection.
Syntax	Remove (Index As Variant)

Method	Remove
Parameters	Index - key to identify the batch object within the collection

2.1.7 ecnImage Object

Description: This object represents a collection of fields within a document.

Properties	Type	Description
Key	String	Returns the name of the image file.

2.1.8 ecnImages Object

Description: This object represents a collection of images within a document.

Properties	Type	Description
Item (Index As Variant)	ecnImage	Returns an ecnImage reference based on the Index specified.
Count	Long	Returns the number of ecnImages in the collection.

Method	Add
Description	Adds a ecnImage reference into the collection.
Syntax	Add (ByVal Key As String, Optional sBefore As String) As ecnImage
Parameters	Key - value will be assigned to ecnBatch.Key property, and will be used as a Key to identify the batch within the collections Before - It is the key of an existing image in the collection, new image object will be added before that image reference.

Method	Clear
Description	Clears out all images within the collection.
Syntax	Clear ()
Parameters	None

Method	Remove
Description	Removes a specific image out of the collection.
Syntax	Remove (Index As Variant)
Parameters	Index - key to identify the image object within the collection

2.1.9 ecnPicklist Object

Description: This object represents a list of valid items for a particular index field.

Properties	Type	Description
Count	Integer	Returns the number of items in the pick-list.

Method	**AddString**
Description	Adds a string to the pick-list.
Syntax	AddString <value>, <display value>
Parameters	<value> and <display value> - values to be passed into the string.

Method	**Clear**
Description	Clears all data in the pick-list.
Syntax	Clear ()
Parameters	None

Method	**GetItem**
Description	Returns the specified pick-list item as a string.
Syntax	GetItem(Picklist Item)
Parameters	Picklist Item - Number representing the pick-list item to return

Method	**RemoveItem**
Description	Removes the specified item from the list.
Syntax	RemoveItem(Picklist item)
Parameters	Picklist Item - Number representing the pick-list item to remove.

2.1.10 ecnPrIndex Object

Description: Represents a profile index field.

Properties	Type	Description
AutoPopulateDate	Integer	Returns the date format type of a field Auto Populated with a date. Values are: 1: yyyy-mm-dd 2: yyyy-mm-dd hh:mm 3: yyyy-mm-dd hh:mm:ss 4: yyyy-mm-dd hh:mm:ss AMPM 5: mm-dd-yyyy 6: mm-dd-yyyy hh:mm 7: mm-dd-yyyy hh:mm:ss 8: mm-dd-yyyy hh:mm:ss AMPM 9: dd-mm-yyyy 10: dd-mm-yyyy hh:mm 11: dd-mm-yyyy hh:mm:ss 12: dd-mm-yyyy hh:mm:ss AMPM 13: mm/dd/yyyy 14: mm/dd/yyyy hh:mm 15: mm/dd/yyyy hh:mm:ss 16: mm/dd/yyyy hh:mm:ss AMPM 17: yyyy/mm/dd 18: yyyy/mm/dd hh:mm 19: yyyy/mm/dd hh:mm:ss 20: yyyy/mm/dd hh:mm:ss AMPM
AutoPopulateDefault	String	Returns the text that the field will be auto populated with for fields with an Auto Populate type of *Default Value*.
AutoPopulateType	Integer	Returns the type of data that the field will be auto populated with. Values are: 0: None 1: Scan Date 2: Index Date 3: Send Date/Time 4: Received Date/Time 5: Default Value 6: Batch Name 7: User ID
DataType	Integer	Returns the data type for the field.
DefaultValue	String	Returns the default value for the field.
FieldLock	Boolean	Returns True if the field cannot be edited.
FieldName	String	Returns the name of the index field.
InputMask	String	Returns the input mask for the field (same as Index).
Key	String	Key value of the field.
Length	Integer	Returns the maximum length of the data allowed in this field.

Properties	Type	Description
MaxValue	Integer	Returns the largest numeric value allowed in the field.
MinValue	Integer	Returns the smallest numeric value allowed in the field.
Required	Boolean	Returns True if this is a required field.
Visible	Boolean	Sets/returns whether a field will be displaying during indexing.

2.1.11 ecnPrIndexes Object

Description: This object represents a collection of indexes within a profile.

Properties	Type	Description
Item (Index As Variant)	EcnIndex	Returns an ecnPrIndex reference based on the Index specified.
Count	Long	Returns the number of ecnPrIndexes in the collection.

Method	Add
Description	Adds an ecnPrIndex reference into the collection.
Syntax	Add (Key As String, Optional sKey As String) As ecnPrIndexes
Parameters	Key - value will be assigned to ecnPrIndex.Key property sKey - value will be used as a key for the collection.

Method	AddIndex
Description	Adds an ecnPrIndex reference into the collection.
Syntax	Add (oIndex as ecnPrIndex, Optional sKey As String)
Parameters	oIndex - a reference of the ecnPrIndex object that will be added into the collections Key - value will be used as a key for the ecnPrIndex object in the collection.

Method	Clear
Description	Clears out all Indexes within the collection.
Syntax	Clear ()
Parameters	None

Method	Remove
Description	Removes a specific ecnPrIndex out of the collection.
Syntax	Remove (Index As Variant)
Parameters	Index - key to identify the ecnPrIndex object within the collection

2.1.12 ecnProfile Object

Description: This object represents a scan profile.

Properties	Type	Description
AutoPopDocIndexed	Boolean	Returns True if the user is allowed to send Auto Populated batches without reviewing them first.
BatchPrefix	String	Returns the string used as a prefix for the batch name.
BlankByteThreshold	Long	Returns the number of bytes used to determine a blank page.
CommitMethod	Long	The type of committing that will take place, where: 0=Do Not Commit, 1=Use Commit Server, 2=Commit Immediately.
DBLookupHitList	Boolean	Returns True if the Hit List for Database Lookups should always be displayed.
DBLookupMaxRecs	Long	Returns the maximum number of records to be returned from a Database Lookup.
DefaultPriority	Integer	Returns the priority number assigned to a batch when it is created.
DefaultStatus	String	Returns the status assigned to a batch when it is created.
Description	String	Returns a description of the scan profile.
DocOption	Integer	Returns the document creation option for the batch, where: 1 = Scan only 2 = Scan and index 3 = Batch Scanning and Indexing
FileCabinet	String	Returns the name of the file cabinet associated with the profile.
ImageFormat	Long	Type of image that will be supported, where: 2=TIFF Group4, 3=TIFF with JPG compression, 4=JPG.
ImageType	Long	Color type, where: 1=B&W low resolution, 2=B&W medium resolution, 3=B&W high resolution, 4=grayscale low resolution, 5=grayscale medium resolution, 6=grayscale high resolution, 7=color low resolution, 8=color medium resolution, and 9=color high resolution.
MaxPages	Integer	Returns the maximum number of pages allowed in batches scanned.
PrIndexes		Collection of PrIndex objects.
ProfileName	String	Returns the name of the profile.
ProfileStatus	Integer	Profile’s status, where 0=inactive and 1=Active.
ScanningType	Long	Scanning type, where 1=scan, 2=scan and index, 3=scan and index batches.
ScriptName	String	Returns the name of the script associated with the profile.
SepByteThreshold	Long	Returns the number of bytes used to determine as separator page.
Statuses		Collection of status objects.

2.1.13 ecnStatus Object

Description: This object represents a status that can be assigned to a batch.

Properties	Type	Description
StatusName	String	Sets/returns the name of the status.
StatusDescription	String	Sets/returns the description of the status.

2.1.14 ecnStatuses Object

Description: This object represents a collection of statuses within a profile.

Properties	Type	Description
Item (Index As Variant)	ecnStatus	Returns an ecnStatus reference based on the Index specified.
Count	Long	Returns the number of ecnStatuses in the collection.

Method	Add
Description	Adds an ecnStatus reference into the collection.
Syntax	Add (ByVal Key As String, Optional sKey As String) As ecnStatus
Parameters	Key - value will be assigned to ecnStatus.Key property, and will be used as a Key to identify the item within the collections Key - Not Used

Method	Clear
Description	Clears out all statuses within the collection.
Syntax	Clear ()
Parameters	None

Method	Remove
Description	Removes a specific status out of the collection.
Syntax	Remove (Index As Variant)
Parameters	Index - key to identify the status object within the collection

2.1.15 ecNet Object

The ecNet object is global and available to all procedures in this module. For information about the Web Capture objects, see "Web Capture Objects" on page 2-1.

Choose from the following properties:

Properties	Description
ActiveBatch	Returns an ecnBatch reference to the batch that was selected by the user (may be nothing).
ActiveDocument	Returns an ecnDocument reference to the document that was selected by the user (may be nothing).

Properties	Description
ActiveImage	Returns an ecnImage reference to the currently selected image.
ActiveProfile	Returns a reference to the currently selected profile.
Batches	Returns an ecnBatches reference containing all batches.
CurrentUserID	Returns the ID of the user that signed into the client.

DisplayMode	Returns an integer representing the state of the display, where:
	ecnSettings = 0
	ecnBatches = 1
	ecnReview = 2
	ecnIndexing = 3

Profiles	Returns an ecnProfiles reference containing all of the scan profiles available. ecnProfiles is a collection of ecnProfile objects.

Method	DeleteBatch
Description	Deletes the currently selected batch. This method can only be used from the batch view.
Syntax	DeleteBatch(Prompt)
Parameters	The Prompt parameter is a Boolean, which indicates if a warning message should be displayed. If true, the user may cancel the delete by answering no at the prompt.

Method	DeletePage
Description	Deletes the currently selected page. May only be called when DisplayMode is ecnReview or ecnIndexing. (See the DisplayMode property described in "ecNet Object" on page 2-10.
Syntax	DeletePage(Prompt)
Parameters	Prompt is a Boolean that indicates if a warning message should be displayed. If True, the user may cancel the deletion by choosing No at the prompt.

Method	ExecDBSearch
Description	Invokes a database search by specifying a search field name, search value and whether to display a hit list showing the results.
Syntax	ExecDBSearch(FieldName, FieldValue, DisplayHitList)
Parameters	This function returns a two dimensional array of search results. The first dimension of the array is for the row, and the second dimension is for the column. The first row in the array contains the field names of the data returned. The actual data starts at row (or subscript) 1. NOTE: DisplayHitList is only valid in ecnIndexing mode.

Method	GetIndexFieldValue
Description	Returns the value of a particular field.
Syntax	GetIndexFieldValue(ByVal FieldName As String) As String
Parameters	FieldName is the name of the field to retrieve the value.

Method	**GetRowIndex**
Description	Returns the current active field.
Syntax	GetRowIndex () As Integer
Parameters	None

Method	**GetSetting**
Description	Retrieves the value of the specified setting.
Syntax	GetSetting(SettingName)
Parameters	SettingName is the setting to be retrieved.

Method	**RefreshCurrentDocument**
Description	Populates the currently displayed document's index fields with data from the active document object.
Syntax	RefreshCurrentDocument()
Parameters	None

Method	**SaveSetting**
Description	Saves a string value for later retrieval.
Syntax	SaveSetting(Setting, Value)
Parameters	Setting is the name given to the setting. Value is the value to save.

Method	**ScanBatch**
Description	Starts scanning a new batch. This method is only valid when the DisplayMode is ecnBatches. (See the DisplayMode property described in "ecNet Object" on page 2-10.
Syntax	ScanBatch()
Parameters	None

Method	**SelectBatch**
Description	Selects the batch in the list that is specified in the BatchName parameter. May only be called when DisplayMode is ecnBatches. (See the DisplayMode property described in "ecNet Object" on page 2-10.
Syntax	SelectBatch(BatchName)
Parameters	BatchName is the batch to be selected.

Method	**SelectPage**
Description	Selects the page specified by the PageNum parameter within the document specified by the DocumentNum parameter. May only be called when DisplayMode is ecnReview or ecnIndexing. (See the DisplayMode property described in "ecNet Object" on page 2-10.
Syntax	SelectPage(PageNum, DocumentNum)

Method	**SelectPage**
Parameters	PageNum is the page to be selected. DocumentNum is an optional parameter, and when specified, indicates the document in which the page resides. If DocumentNum is not specified, the current document is assumed. NOTE: If PageNum is less than 1, the document itself will be selected.

Method	**SetDisplayMode**
Description	Sets the current display to one of the valid display modes.
Syntax	SetDisplayMode(DisplayMode)
Parameters	DisplayMode is the mode to be set. (See the DisplayMode property described in "ecNet Object" on page 2-10.

Method	**SetIndexFieldValue**
Description	Sets the value of a particular field.
Syntax	SetIndexFieldValue(ByVal FieldName As String, ByVal FieldValue As String)
Parameters	None

Method	**SetRowIndex**
Description	Sets the current active field.
Syntax	SetRowIndex (ByVal RowIndex As Integer)
Parameters	RowIndex is the row of the field which needs focus.

Method	**SetStatusCaption**
Description	Sets the caption text in the status bar.
Syntax	SetStatusCaption(CaptionText)
Parameters	CaptionText is the status text to be set.

2.2 Web Capture Client Events

Choose from the following events.

Event	**AppendBegin**
Description	Occurs when the user chooses to append a page.
Syntax	ecNet_AppendBegin(Cancel)
Parameters	Set Cancel to true to cancel the page appending.

Event	**AppendComplete**
Description	Occurs after a page has been appended or after the user has canceled the append.

Event	AppendComplete
Syntax	ecNet_AppendComplete(ByVal Cancel, ByVal PagesAppended, ByVal ErrorNum, ByVal ErrorDesc)
Parameters	Cancel is set to True if the user canceled the append. PagesAppended contains the number of pages that were appended. If an error occurred, ErrorNum contains a number and ErrorDesc contains a description that identify the error.

Event	BatchDelete
Description	Occurs when a batch is about to be deleted.
Syntax	ecNet_BatchDelete(ByVal Batch, Cancel)
Parameters	Batch is the batch object that is being referenced. Set Cancel to true to cancel the delete.

Event	BatchDeleteComplete
Description	Occurs after a batch has been deleted or after the user canceled the batch deletion.
Syntax	ecNet_BatchDeleteComplete(ByVal Deleted, ByVal ErrorNum, ByVal ErrorDesc)
Parameters	Deleted is set to True if the batch was deleted. If an error occurred, ErrorNum contains a number and ErrorDesc contains a description that identify the error.

Event	BatchImportBegin
Description	Occurs when importing into a batch is about to begin.
Syntax	ecNet_BatchImportBegin(ByVal Batch)
Parameters	Batch is the batch object that is being referenced.

Event	BatchImportComplete
Description	Occurs when importing into a batch is complete.
Syntax	ecNet_BatchImportComplete(ByVal Batch)
Parameters	Batch is the batch object that is being referenced.

Event	BatchPostDelete
Description	Occurs after all selected batches have been deleted, regardless of whether the user canceled the deletion.
Syntax	ecNet_BatchPostDelete(ByVal Cancel)
Parameters	Cancel is set to True if the user canceled the deletion.

Event	BatchPreDelete
Description	Occurs before all selected batches are about to be deleted.
Syntax	ecNet_BatchPreDelete(Cancel)

Event	BatchPreDelete
Parameters	Set Cancel to true to cancel the deletion.

Event	BatchReviewClose
Description	Occurs just before a batch is closed in the batch review window.
Syntax	ecNet_BatchReviewClose(ByVal Batch, Cancel)
Parameters	Batch is the batch object that is being referenced. Set Cancel to true to cancel closing the batch review window.

Event	BatchReviewOpen
Description	Occurs just before a batch is open for review.
Syntax	ecNet_BatchReviewOpen(ByVal Batch, Cancel)
Parameters	Batch is the batch object that is being referenced. Set Cancel to true to cancel the review.

Event	BatchScanBegin
Description	Occurs when scanning into a batch is about to begin.
Syntax	ecNet_BatchScanBegin(ByVal Batch)
Parameters	Batch is the batch object that is being referenced.

Event	BatchScanComplete
Description	Occurs when scanning into a batch is complete.
Syntax	ecNet_BatchScanComplete(ByVal Batch)
Parameters	Batch is the batch object that is being referenced.

Event	DBSearchComplete
Description	Occurs when the search has completed, just before the results are to be processed.
Syntax	ecNet_DBSearchComplete(Byval Results, Cancel)
Parameters	Results is a two dimensional array (row, col) containing the results. When Cancel is set to True, the results window is not displayed, the search is canceled, the results are ignored and no mapped fields are populated.

Event	DBSearchResult
Description	Occurs as search results are being processed from the recordset, before the results are displayed.
Syntax	ecNet_DBSearchResult(Byval FieldName, FieldValue)
Parameters	FieldName is the name of the result field. FieldValue is the value of the result field.

Event	DBSearchStart
Description	Occurs just before the search.
Syntax	ecNet_DBSearchStart(FieldName, FieldValue, Cancel)
Parameters	FieldName is the name of the field that will be used to search on. FieldValue is the value being sought.

Event	DeleteDocumentBegin
Description	Occurs when the user chooses to delete a document.
Syntax	ecNet_DeleteDocumentBegin(Cancel)
Parameters	Set Cancel to True to cancel the deletion.

Event	DeleteDocumentComplete
Description	Occurs after a document has been deleted or after the user has canceled the document deletion.
Syntax	ecNet_DeleteDocumentComplete(ByVal Cancel, ByVal ErrorNum, ByVal ErrorDesc)
Parameters	Cancel is set to True if the user canceled the document deletion. If an error occurred, ErrorNum contains a number and ErrorDesc contains a description that identify the error.

Event	DeletePageBegin
Description	Occurs when the user chooses to delete a page.
Syntax	ecNet_DeletePageBegin(Cancel)
Parameters	Set Cancel to true to cancel the page deletion.

Event	DeletePageComplete
Description	Occurs after a page has been deleted or after the user canceled the page deletion.
Syntax	ecNet_DeletePageComplete(ByVal Cancel, ByVal ErrorNum, ByVal ErrorDesc)
Parameters	Cancel is set to True if the user canceled the page deletion. If an error occurred, ErrorNum contains a number and ErrorDesc contains a description that identify the error.

Event	DocumentCreated
Description	Occurs after a document has been created.
Syntax	ecNet_DocumentCreated()
Parameters	None

Event	DocumentNext
Description	Occurs before a move to the next document.

Event	DocumentNext
Syntax	ecNet_DocumentNext(Cancel)
Parameters	Set Cancel to true to prevent the document change.

Event	DocumentOnCreate
Description	Occurs when the user is about to create a new document.
Syntax	ecNet_DocumentOnCreate(Cancel)
Parameters	Set Cancel to true to prevent the document from being created.

Event	DocumentOnRemove
Description	Occurs when the user is about to delete a document.
Syntax	ecNet_DocumentOnRemove(Cancel)
Parameters	Set Cancel to true to prevent the document from being deleted.

Event	DocumentPostProcess
Description	Occurs after document data has been added to the pak file.
Syntax	ecNet_DocumentPostProcess(Document)
Parameters	Document is the document being processed.

Event	DocumentPreProcess
Description	Occurs before document data is about to be added to the pak file.
Syntax	ecNet_DocumentPreProcess(PakFilename, Document)
Parameters	FileName is the name of the pak file that the document will be added to. Document is the document being processed.

Event	DocumentPrevious
Description	Occurs before a move to a previous document.
Syntax	ecNet_DocumentPrevious(Cancel)
Parameters	Set Cancel to true to prevent the document change.

Event	DocumentRemoved
Description	Occurs after the document has been removed.
Syntax	ecNet_DocumentRemoved()
Parameters	None

Event	DocumentSelected
Description	Occurs when the user selects a document in the document tree.
Syntax	ecNet_DocumentSelected(Document)

Event	DocumentSelected
Parameters	Document is the document object representing the document that the user clicked on in the tree.

Event	FieldGotFocus
Description	Occurs when an indexing field receives the input focus.
Syntax	ecNet_FieldGotFocus(ByVal FieldName)
Parameters	FieldName is the name of the field entering focus.

Event	FieldLostFocus
Description	Occurs when a field is about to lose the input focus.
Syntax	ecNet_FieldLostFocus(ByVal FieldName, Cancel)
Parameters	FieldName is the name of the field losing focus. If Cancel is set to true, will force the user to remain on the current field.

Event	FieldKeyDown
Description	Occurs when the user presses down on a key when focus is in a data field.
Syntax	ecNet_FieldKeyDown(Byval KeyCode, Byval Shift)
Parameters	Keycode is an integer representing the key pressed on the keyboard. For details, see "Keycodes" on page A-1. Shift is an integer representing the status of the Ctrl, Alt and Shift keys.

Event	FieldKeyPress
Description	Occurs when a key has been pressed in an input field.
Syntax	ecNet_FieldKeyPress(KeyAscii)
Parameters	KeyAscii is the ASCII value for the key pressed.

Event	FieldKeyUp
Description	Occurs when the user releases a key when focus is in a data field.
Syntax	ecNet_FieldKeyUp(Byval KeyCode, Byval Shift)
Parameters	Keycode is an integer representing the key pressed on the keyboard. For details, see "Keycodes" on page A-1. Shift is an integer representing the status of the Ctrl, Alt and Shift keys.

Event	ImportFilesSelected
Description	Occurs after a list of files is selected for import.
Syntax	ecNet_ImportFilesSelected(Files, Cancel)
Parameters	ecNet_ImportFilesSelected(Files, Cancel)File is a collection of file names ready for importing. Cancel set to true cancels importing.

Event	InsertBegin
Description	Occurs when the user chooses to insert a page.
Syntax	ecNet_InsertBegin(Cancel)
Parameters	Cancel set to true cancels the page insertion.

Event	InsertComplete
Description	Occurs after a page has been inserted or after the user has canceled the insert.
Syntax	ecNet_InsertComplete(ByVal Cancel, ByVal PagesInserted, ByVal ErrorNum, ByVal ErrorDesc)
Parameters	Cancel is set to True if the user canceled the insert. PagesInserted contains the number of pages inserted. If an error occurred, ErrorNum contains a number and ErrorDesc contains a description that identify the error.

Event	PageNext
Description	Occurs when the user clicks the next page button.
Syntax	ecNet_PageNext(Cancel)
Parameters	Set Cancel to true to prevent the user from going to the next page.

Event	PagePrevious
Description	Occurs when the user clicks the previous page button.
Syntax	ecNet_PagePrevious(Cancel)
Parameters	Set Cancel to true to prevent the user from going to the previous page.

Event	ReplaceBegin
Description	Occurs when the user chooses to replace a page.
Syntax	ecNet_ReplaceBegin(Cancel)
Parameters	Set Cancel to true to cancel the replacement.

Event	ReplaceComplete
Description	Occurs after a page has been replaced or after the user has canceled the replacement.
Syntax	ecNet_ReplaceComplete(ByVal Cancel, ByVal ErrorNum, ByVal ErrorDesc)
Parameters	Cancel is set to True if the user canceled the replacement. If an error occurred, ErrorNum contains a number and ErrorDesc contains a description that identify the error.

Event	ScriptStart
Description	This is the first event fired when the script starts.
Syntax	ecNet_ScriptStart()

Event	**ScriptStart**
Parameters	None

Event	**SelectBatch**
Description	Occurs when a batch in the batch list is selected.
Syntax	ecNet_SelectBatch(Byval Batch)
Parameters	Batch is the batch selected in the batch list.

Event	**SelectProfile**
Description	Occurs when a new profile is selected in the UI.
Syntax	ecNet_SelectProfile(Byval Profile)
Parameters	Profile is the profile that was selected.

Event	**SendBegin**
Description	Occurs when a batch is about to send across the Internet.
Syntax	ecNet_SendBegin(Cancel)
Parameters	Set Cancel to true to abort the send.

Event	**SendComplete**
Description	Occurs when the batch send has completed.
Syntax	ecNet_SendComplete()
Parameters	None

Event	**SendError**
Description	Occurs if a batch fails to send.
Syntax	ecNet_SendError(Batch, ErrorNum, ErrorDesc, Cancel)
Parameters	*Batch* is the batch that failed to send. *ErrorNum* is the error number that identifies the error. *ErrorDesc* is a description of the error. *Cancel* set to true cancels any remaining send operations; by default, Cancel is false.

A

Keycodes

Keyboard keycodes A - Z are the same as their ASCII equivalents:

Keycode	Key	Keycode	Key
65	A key	78	N key
66	B key	79	O key
67	C key	80	P key
68	D key	81	Q key
69	E key	82	R key
70	F key	83	S key
71	G key	84	Y key
72	H key	85	U key
73	I key	86	V key
74	J key	87	W key
75	K key	88	X key
76	L key	89	Y key
77	M key	90	Z key

Keyboard keycodes 0 - 9 are the same as their ASCII equivalents:

Keycode	Key	Keycode	Key
48	0 key	53	5 key
49	1 key	54	6 key
50	2 key	55	7 key
51	3 key	56	8 key
52	4 key	57	9 key

Keys on the Numeric Keypad:

Keycode	Key	Keycode	Key
96	0 key	104	8 key
97	1 key	105	9 key

Keycode	Key	Keycode	Key
98	2 key	106	MULTIPLICATION SIGN (*) key
99	3 key	107	PLUS SIGN (+) key
100	4 key	13	ENTER key
101	5 key	109	MINUS SIGN (-) key
102	6 key	110	DECIMAL POINT (.) key
103	7 key	111	DIVISION SIGN (/) key

Function Keys:

Keycode	Key	Keycode	Key
112	F1 key	120	F9 key
113	F2 key	121	F10 key
114	F3 key	122	F11 key
115	F4 key	123	F12 key
116	F5 Key	124	F13 key
117	F6 key	125	F14 key
118	F7 key	126	F15 key
119	F8 key	127	F16 key

Miscellaneous Keys:

Keycode	Key	Keycode	Key
1	Left mouse button	34	PAGE DOWN key
2	Right mouse button	35	END key
3	CANCEL key	36	HOME key
4	Middle mouse button	37	LEFT ARROW key
8	BACKSPACE key	38	UP ARROW key
9	TAB key	39	RIGHT ARROW key
12	CLEAR key	40	DOWN ARROW key
13	ENTER key	41	SELECT key
16	SHIFT key	42	PRINT SCREEN key
17	CTRL key	43	EXECUTE key
18	MENU key	44	SNAPSHOT key
19	PAUSE key	45	INS key
20	CAPS LOCK key	46	DEL key
27	ESC key	47	HELP key
32	SPACEBAR key	144	NUM LOCK key
33	PAGE UP key		

B

Copyright and Patent Notices

Index