dcUTIL_DotNet.dll

Script utility collection for use with Kofax Capture

v3.2

Contents

1 Overview

dcUTIL_DotNet.dll is a collection of utilities developed for use within Kofax Capture and KTM. These are primarily designed to be used within Validation scripts to enhance the standard functionality available within Kofax. However, as they are independent external components, they can also be used elsewhere – for example within Recognition.

The utilities vary from simple text file lookups, to more advanced SQL searches and automatic dictionary-based text correction utilities. These calls are detailed below.

Whenever database type utilities are called, connections are made via a System DSN allowing greater flexibility to change or amend functionality without heavy script editing.

1.1 New Features and Enhancements

1.1.1 3.0

Increased Support	All functionality is now supported in SBL, VB.Net and KTM scripting.
Column Reorder	All display screens now have the ability to reorder columns to speed up result selection.
Display Resize	The display window can be resized.
Error Logging	A log file location can be specified for error logging.
Installer	An installer utility allows automatic registering of components.
Match Two Fields	A new function call allows AND/OR searches to be run against 2 columns.
FileBrowser	Call a file browser from within a script to select a filename.
FolderBrowser	Call a folder browser from within a script to select a folder.
SpellChecker	Call an interactive spell checker from with a script.
flexSEARCH	Call a completely interactive and flexible lookup from within a script.

1.1.2 3.1

Installation	Installation and uninstallation now has an improved interface and all files are removed during uninstall.
MySQL	dcUTIL_DotNet now support MySQL for all database functions.

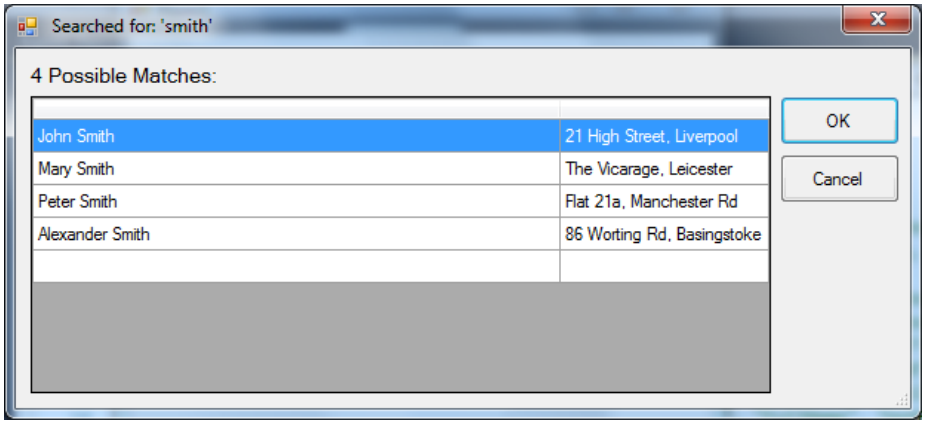
2 Utility Calls within dcUTIL

2.1 dbMATCH

This call is used to search within a database table/column and return a corresponding value from a second column within the table.

Where a single match is found, this is passed automatically to the calling code.

Where multiple matches are found, a list is displayed for a user to select the appropriate value.

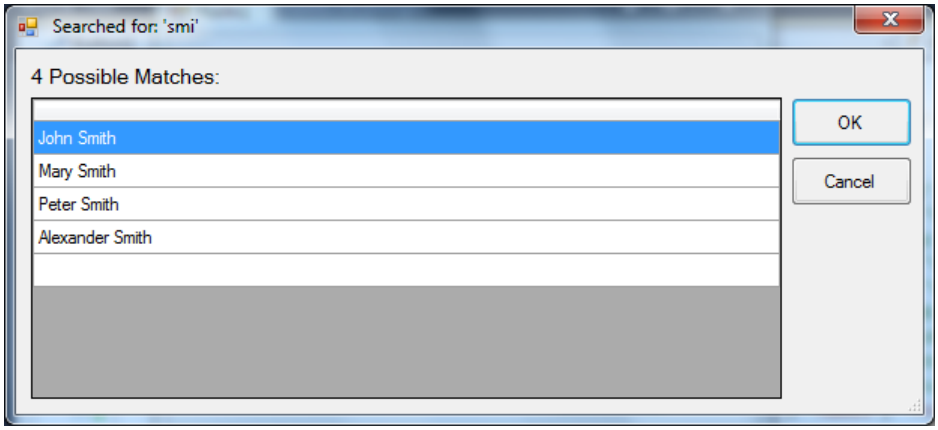


2.2 dbLOOKUP

This call is used to search within a database table/column and return a value based on a search string entered. The search can be a 'begins with' or 'contains' search.

Where a single match is found, this is passed automatically to the calling code.

Where multiple matches are found, a list is displayed for a user to select the appropriate value.



2.3 isEXVAL

This call simply checks within a database table/column to see if the value passed exists in the table. This can be used simply as a validation, or to determine whether a further lookup is required.

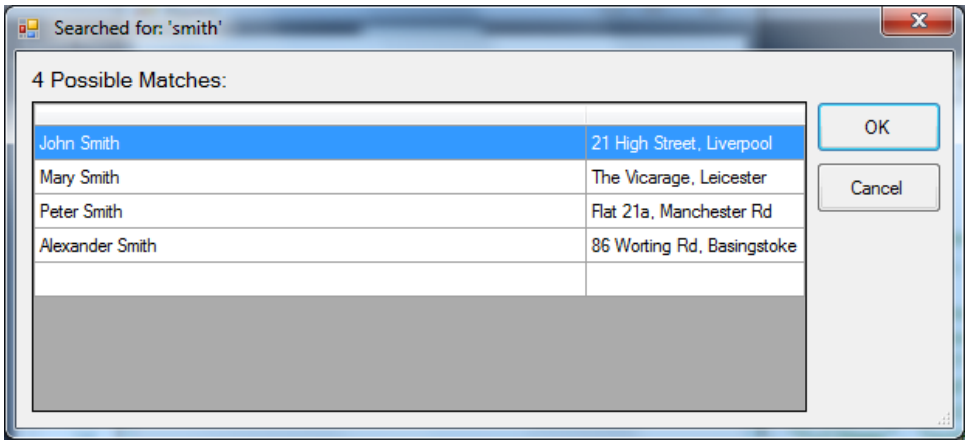
The return is TRUE/FALSE and no data is passed back.

2.4 dbMATCHMULT

Function to do a lookup into database and match a value passed and return up to three columns from the resulting matched row.

Where a single match is found, this is passed automatically to the calling code.

Where multiple matches are found, a list is displayed for a user to select the appropriate value.



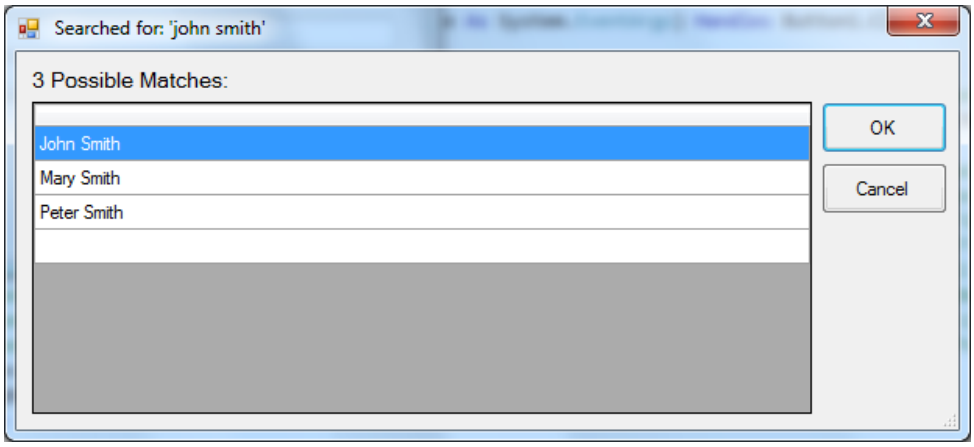
2.5 dbCORRECT

Function to allow the autocorrecting, or matching of string values within a database table and column.

A search string is passed, and the type of search is specified – first match, best match, all matches – and a confidence level to meet is also specified.

Where first or best match is selected, there will be a maximum of one returned string. This can be used to auto-correct the data.

Where all is selected as the search type, a selection box will be shown where there are multiple matches. If only one match, this will be passed.



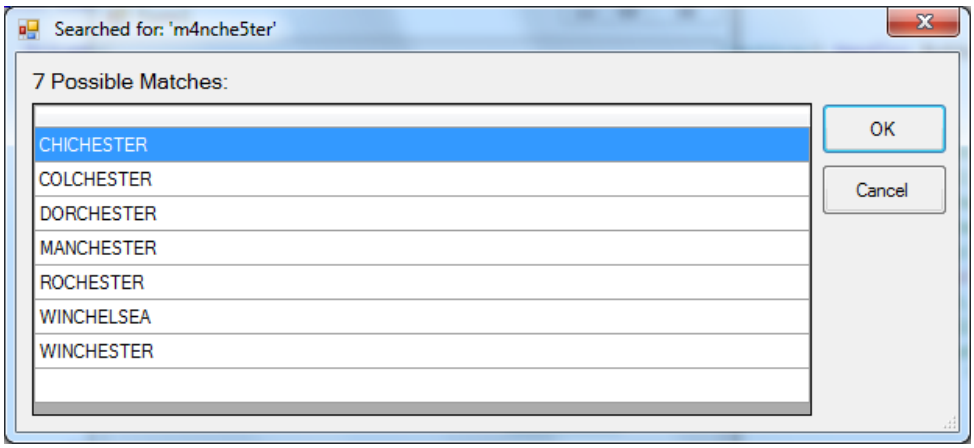
2.6 txtCORRECT

Function to allow the autocorrecting, or matching of string values within a text file.

A search string is passed, and the type of search is specified – first match, best match, all matches – and a confidence level to meet is also specified.

Where first or best match is selected, there will be a maximum of one returned string. This can be used to auto-correct the data.

Where all is selected as the search type, a selection box will be shown where there are multiple matches. If only one match, this will be passed.

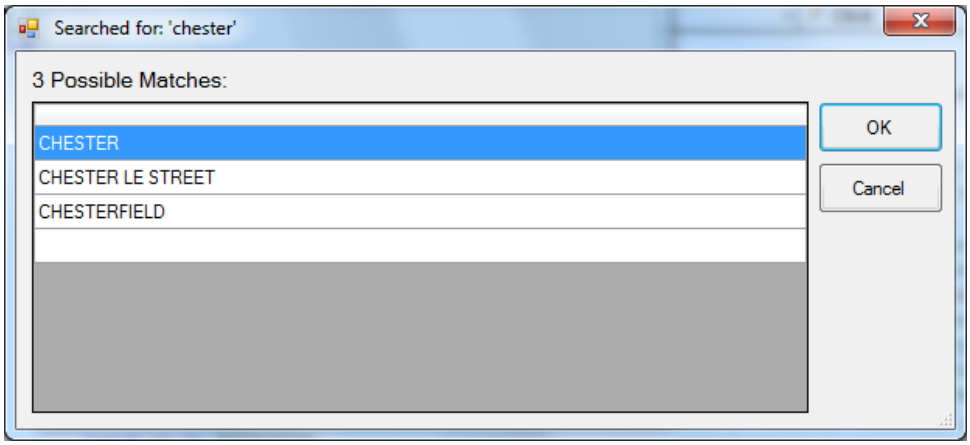


2.7 txtLOOKUP

This call is used to search within a text file and return a value based on a search string entered. The search can be a 'begins with' or 'contains' search.

Where a single match is found, this is passed automatically to the calling code.

Where multiple matches are found, a list is displayed for a user to select the appropriate value.

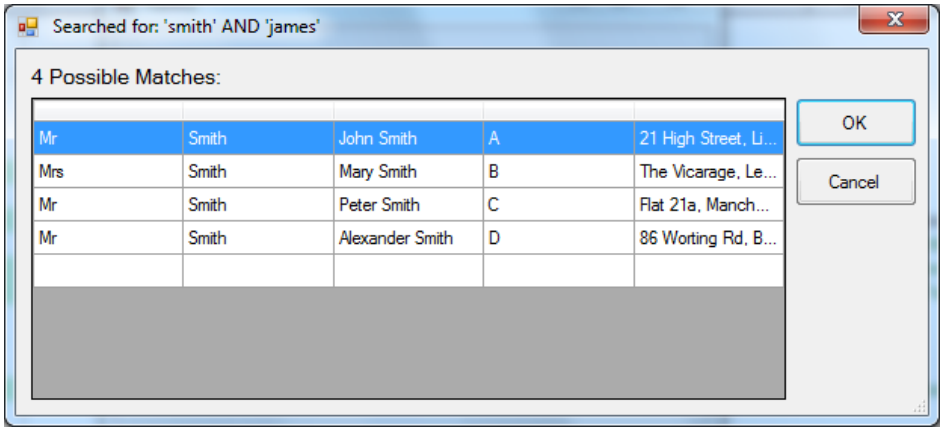


2.8 dbMATCH_Two_Fields

This call is used to match 2 fields within a database (Surname and DateOfBirth for example). The match can be AND/OR to allow greater flexibility, and each criteria can be Equal | Not Equal | Begins | Contains.

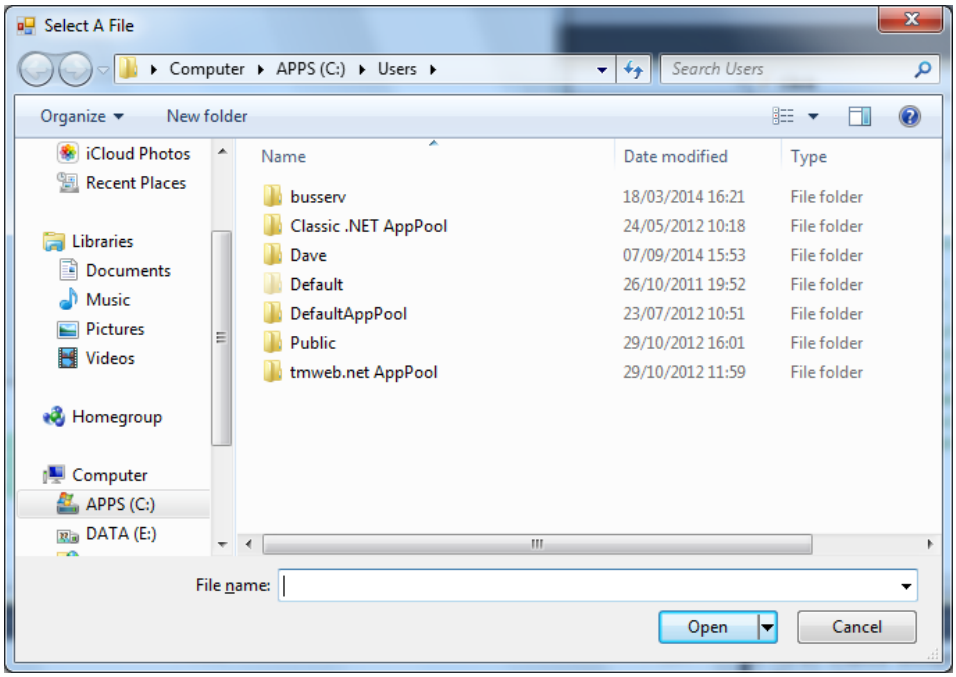
Where a single match is found, the value is returned automatically to the calling code.

Where multiple matches are found, a list is displayed for a user to select the appropriate value.



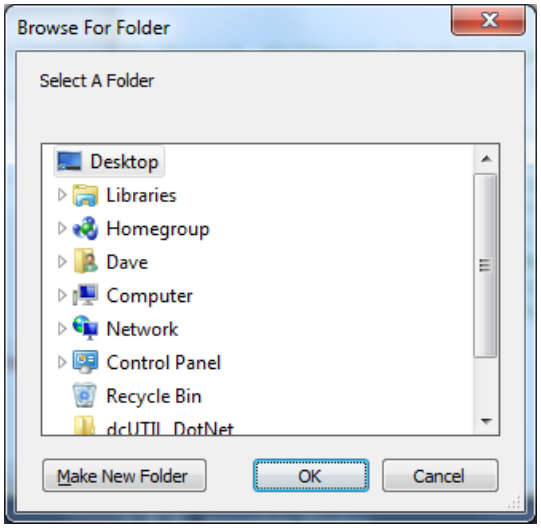
2.9 fileBrowser

This function is used to trigger a standard Windows File Browser. This will allow a user to browse and select a file, and return either the full path or the filename.



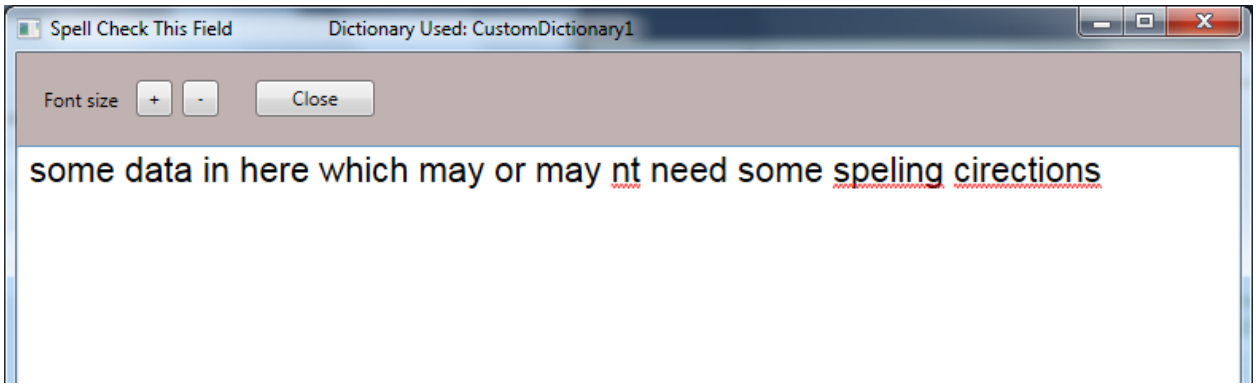
2.10 folderBrowser

This function is used to trigger a standard Windows Folder Browser. This will allow a user to browse (and optionally create) folders, and select one.



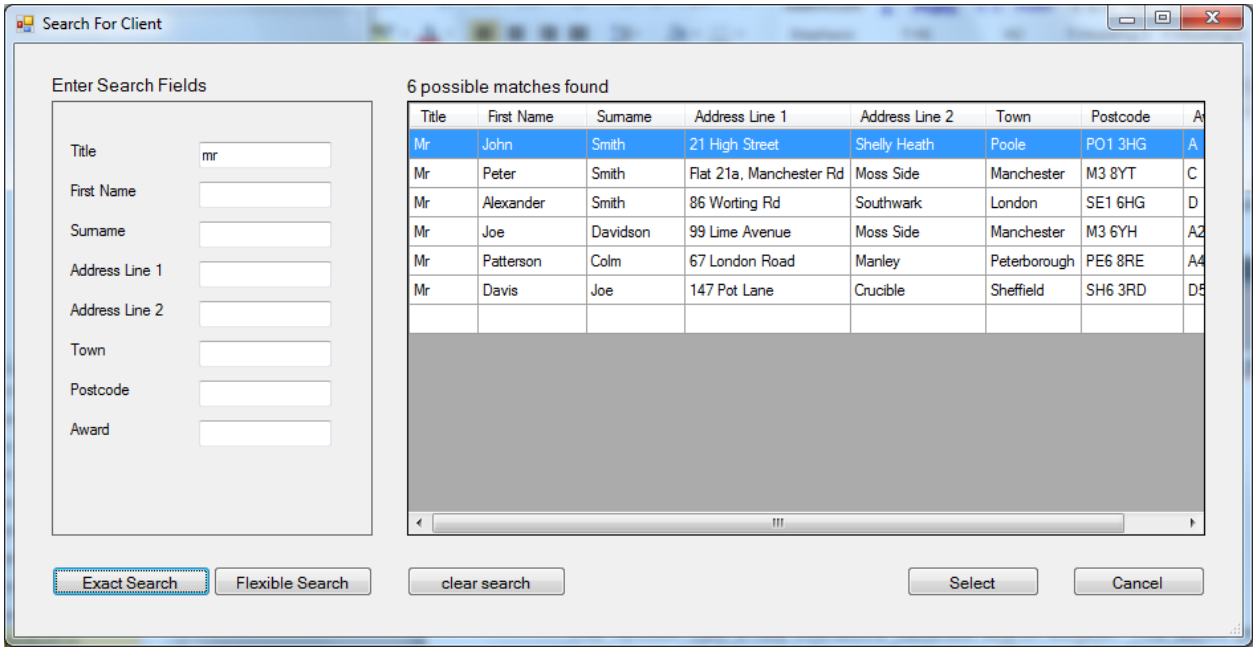
2.11 spellChecker

This function is used to pass text to a spell checking utility. If no mistakes are found, no action is taken. If spelling errors are detected, a window will be displayed showing the words requiring checking.



2.12 flexSEARCH

This function calls a fully interactive database search window. This allows you to point to a database and determine the search columns. These will be shown as search fields, and the user can enter any or all of them and then do either an 'exact' search or a 'flexible' search. The results are displayed for the user who can then choose the appropriate one.



3 Installation and Implementation

In order to use the utilities within dcUTIL.dll, the dll must be installed and registered on each machine where the calling script will be run – usually the Kofax Validation Workstations.

If database utilities are being called, a System DSN must also be configured on each of these stations.

3.1 Upgrading from dcUTIL.dll

If you currently use dcUTIL.dll then follow the installation instructions in 3.2 below.

Once you have installed dcUTIL_DotNet.dll, you simply need to change references in any existing scripts from ***dcUTIL.dll*** to ***dcUTIL_DotNet.dll***

Usually this would only involve changing the entry in the LoadValidation event in a SBL Validation script:

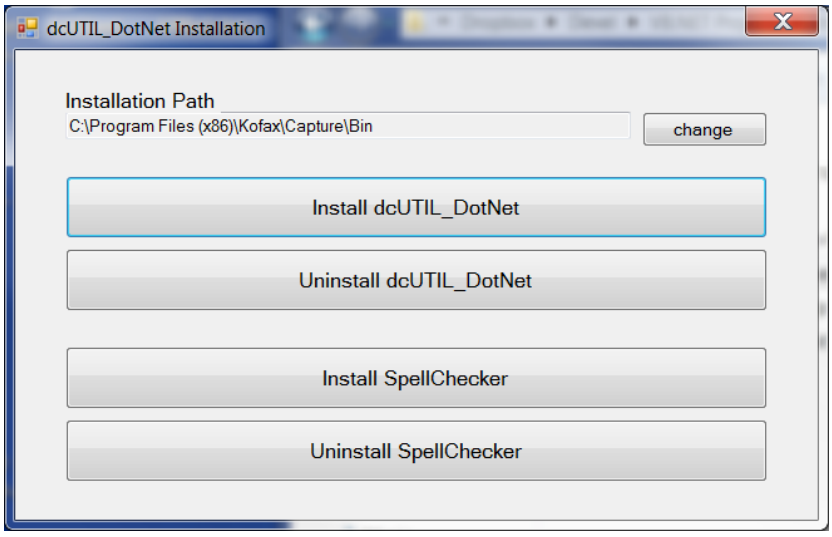
Old **Set dcUTILObject=CreateObject("dcUTIL.autoCORRECT")**
New **Set dcUTILObject=CreateObject("dcUTIL_DotNet.autoCORRECT")**

All function calls from the earlier version are supported without change in the latest versions, so you do not need to make further changes to the scripts.

3.2 Install/Uninstall dcUTIL_DotNet.dll

The steps to install dcUTIL_DotNet.dll are as follows:

1. Unzip the dcUTIL_DotNet.zip folder, and run the **dcUTIL_DotNet_Installer.exe**



2. Select the correct installation folder – normally this should be the Kofax\Capture\Bin folder.

3. Install dcUTIL_DotNet

4. If required, Install SpellChecker (requires .NET 4)

5. Repeat on each workstation which will require dcUTIL_DotNet functionality.

You may require Administrative privileges to register dcUTIL_DotNet.dll and SpellChecker on the workstations. Please check with your IT Administrator if you experience any problems registering.

3.3 Create a System DSN (only required for database utilities)

Create a System DSN using the Data Sources (ODBC) manager.

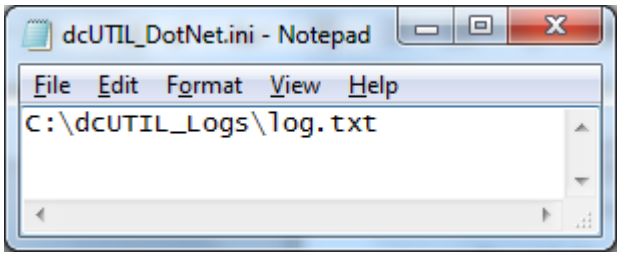
START > CONTROL PANEL > ADMINISTRATIVE TOOLS > Data Sources (ODBC)

Settings and permissions will vary depending on database type and user permissions. Please refer to your System Administrator for assistance.

3.4 Logging

By default dcUTIL_DotNet.dll will log errors encountered to a LOG file located in the installation folder - dcUTIL_DotNet_LOG.txt

If you wish to use a specific log file, then edit the dcUTIL_DotNet.ini file in the installation path to point to a custom log file location.



4 Incorporate into Script

4.1 SBL Scripts

Once the dcUTIL_DotNet.dll has been registered, and a System DSN created (if required), you can incorporate the functionality into a script.

As Kofax Capture Validation Scripts are the most common use for dcUTIL_DotNet.dll, the example below shows how to incorporate the dcUTIL_DotNet.dll functionality within this type of script.

The dcUTIL_DotNet.dll can be used in other scripts and the procedure is largely the same:

1. Declare an object
2. Create instance of dcUTIL_DotNet.dll
3. Call any of the utility functions

Declaration of Object in Kofax Validation Script

At the top of the SBL script, add a declaration for the object to be used later in the script.

```
REM ============================================================================
REM Index fields processed by pre, post, or format procedures must be
REM defined above before any of the functions that actually use them.
REM ----------------------------------------------------------------------------
```

Dim dcUTILObject as Object

```
REM ============================================================================
REM Function handling initialization for this module.
REM This function is called after the user opens a batch.  The function is
REM called once per batch and is called and before any other function in
REM this module.
REM ----------------------------------------------------------------------------
```

Create Instance of Object in LoadValidation Function

```
Function KfxLoadValidation ( VerifyBatch As Integer, NumberOfDocsInBatch As Integer ) As Integer
        On Error GoTo Failure

If (VerifyBatch <> 0) Then
        KfxOperation = "Verify"
Else
        KfxOperation = "Index"
End If
```

Set dcUTILObject=CreateObject("dcUTIL_DotNet.autoCORRECT")

```
KfxLoadValidation = NoError

Exit Function

Failure:
  KfxLoadValidation = FatalError
  Exit Function
End Function
```

Call One of the Utility Functions

Typically the functionality is called from a PostField event in the Validation Script, but it can also be called from any other event. The example below shows the use of the dbMATCH function in a PostField event.

Note: For a Validation field called 'UserAddress' there will be a PostField event function called 'PostUserAddress' and the variable used will be 'KfxUserAddress'.

Please refer to the Kofax Capture help for more information on the structure and flow of Validation Scripts.

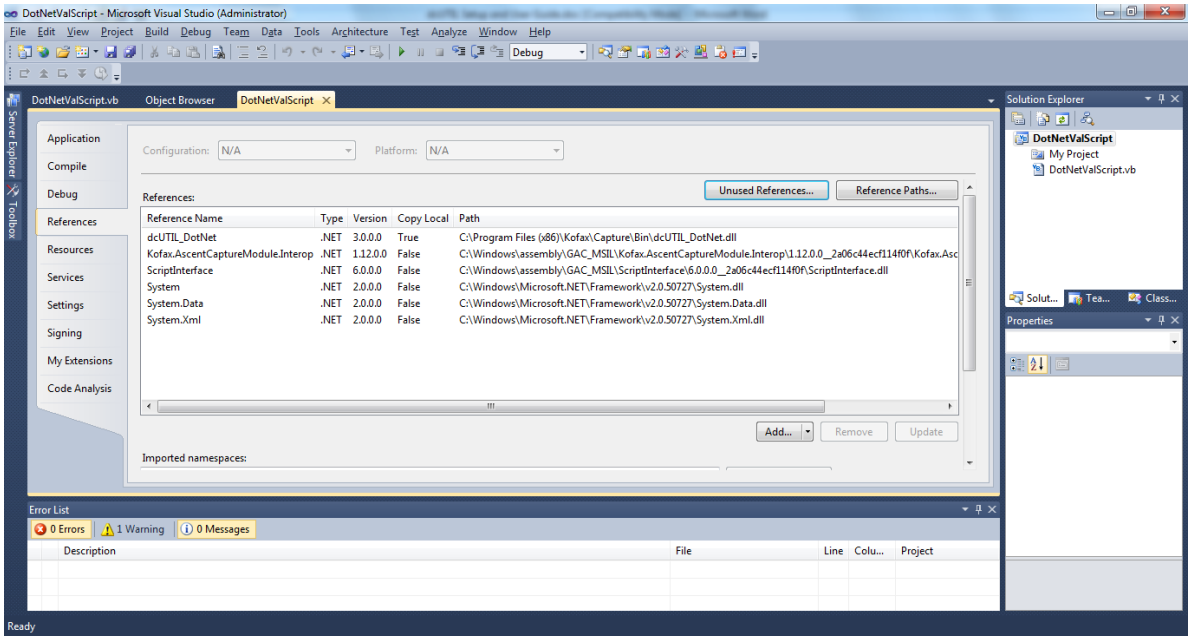
```
Function PostUserAddress( EnteredValue As String, MaxLength As Integer ) As Integer
        On Error GoTo Failure

EnteredValue = Trim(EnteredValue)
If ( Len(EnteredValue) > MaxLength ) Then GoTo Failure
KfxUserAddress= EnteredValue


KfxUserAddress=dcUTILObject.dbMATCH("dsn","table","FullName","Address",KfxUserAddress,1)


PostUserName = NoError
Exit Function

Failure:
  PostUserName = ValidationError
  Exit Function
End Function
```

In the above example the following would happen:

1. User enters a name or partial name in the UserAddress field and hits TAB/RTN to exit field.
2. PostUserAddress event is fired.
3. KfxUserAddress variable is set to the name entered.
4. dbMATCH looks in 'table' via 'dsn' and searches the 'FullName' column for the value in KfxUserAddress. Where value(s) are matched, the 'Address' value is returned and set in KfxUserAddress.

4.2 VB.Net Scripts

Add a reference to dcUTIL_DotNet.dll in the Project References



In the relevant event, add code to call a function.

```
Private Sub OCR_FieldPostProcessing(sender As Object, e As
Kofax.AscentCapture.Scripting.PostFieldEventArgs) Handles OCR.FieldPostProcessing

            'create an instance of the dcUTIL_DotNet utility
            Dim dcUTIL As New dcUTIL_DotNet.autoCORRECT

            'call the SpellChecker function
            OCR.IndexField.Value = dcUTIL.spellChecker(OCR.IndexField.Value,
"D:\Data\CustomDictionary\OCRValues.lex", "Please Check The Spelling in the Selected
Text", True)

End Sub
```

In the example above, the entered value in the OCR Index Field is passed to the SpellChecker, using a custom dictionary.

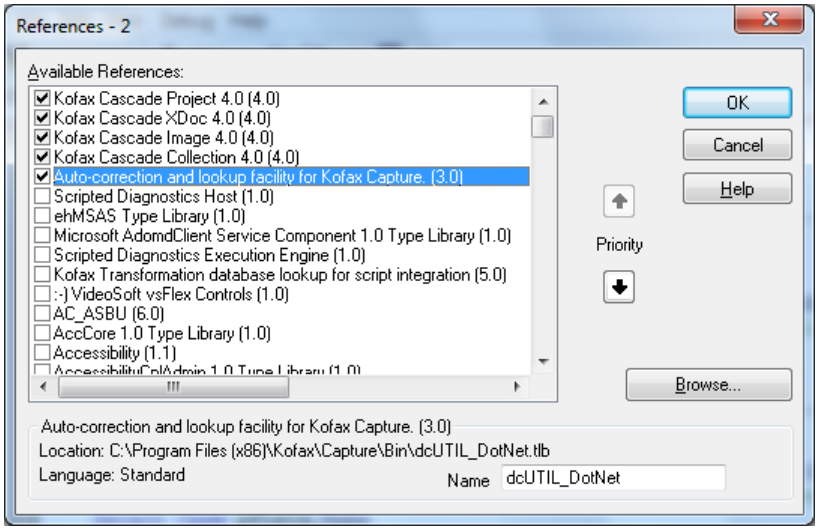
4.3 KTM Scripts

Add a reference to dcUTIL_DotNet.dll in the project script.

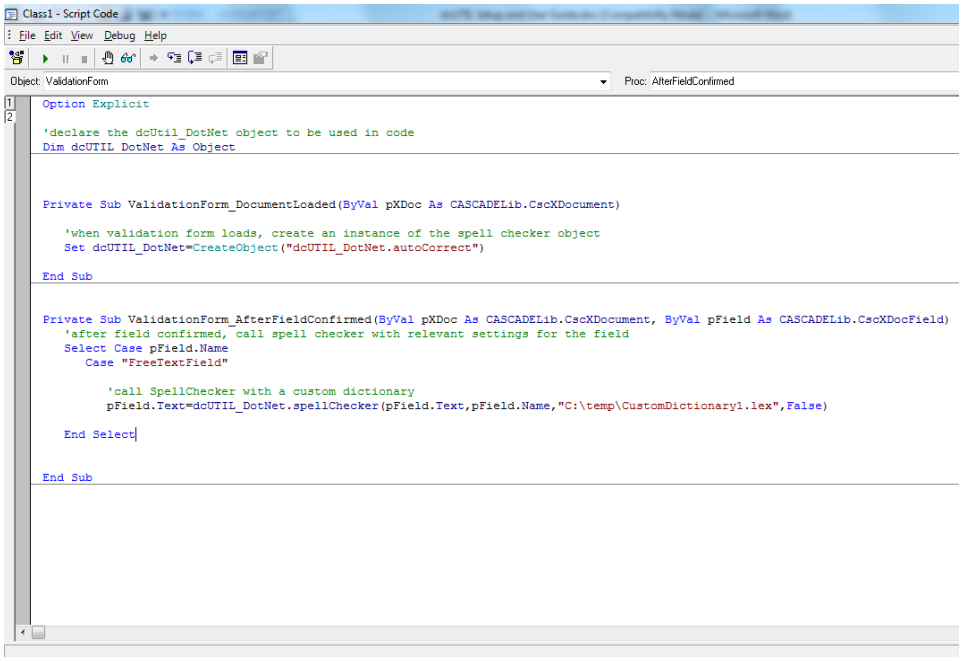
Edit > References

Browse to the dcUTIL_DotNet.tlb file in the installation location, and select it.

NB. Select the **dcUTIL_DotNet.tlb** file, not the dcUTIL_DotNet.dll file.



Add code to the script to create an instance of dcUTIL_DotNet.dll, and call functions.



In the example above the following relevant code is used:

1. Create an object to be used later in the code.

Dim dcUTIL_DotNet As Object

2. Add code to the ValidationForm_DocumentLoaded event to create an instance of dcUTIL_DotNet when the document is loaded in KTM Validation.

Private Sub ValidationForm_DocumentLoaded(ByVal pXDoc As CASCADELib.CscXDocument)

'when validation form loads, create an instance of the spell checker object
Set dcUTIL_DotNet=CreateObject("dcUTIL_DotNet.autoCorrect")

End Sub

3. Call the relevant functions from the created object.

Private Sub ValidationForm_AfterFieldConfirmed(ByVal pXDoc As CASCADELib.CscXDocument, ByVal pField As CASCADELib.CscXDocField)

Select Case pField.Name

Case "FreeTextField"

'call SpellChecker with a custom dictionary
pField.Text=dcUTIL_DotNet.spellChecker(pField.Text,pField.Name,"C:\temp\CustomDictionary1.lex",False)

End Select

End Sub

4.4 Using MySQL

When using MySQL a property must be set on the main dcUTIL_DoNet object used in script.

For example:

Set dcUTIL_DotNet=CreateObject("dcUTIL_DotNet.autoCorrect")
dcUTIL_DotNet.isMySql=True

5 Function Call Details

5.1 dbMATCH

5.1.1 Overview

```
Function dbMATCH(    dsn_NAME As String
                     table_NAME As String
                     search_COLUMN As String
                     return_COLUMN As String
                     search_VALUE As String
                     show_ALL As Short) As String
```

This function is passed a set of parameters and does a lookup within a database. The lookup searches a column within a table and returns the corresponding entry from a second column within the table.

Where there is a single match, the value is passed automatically. Where there are multiple matches, a selection list is displayed for a manual selection.

Note: the search uses LIKE in the SQL call so does not require a case sensitive or exact match.

5.1.2 Parameters

dsn_NAME	the dsn to use to connect to database
table_NAME	the table within database to use
search_COLUMN	the column to search
return_COLUMN	the column to return
search_VALUE	the string to search for
show_ALL	flag to determine whether to show return_COLUMN value 0 – search_COLUMN 1 – search_COLUMN | return_COLUMN

5.1.3 Returns

If a match found	the matching string
If no match found	“”
If error	**Error description**

5.2 dbLOOKUP

5.2.1 Overview

This function is passed a set of parameters and does a search within a database/column. A flag is set to determine whether the lookup uses a 'begins with' or 'contains' search.

Where there is a single match, the value is passed automatically. Where there are multiple matches, a selection list is displayed for a manual selection.

```
Function dbLOOKUP(   dsn_NAME As String
                     table_NAME As String
                     column_NAME As String
                     search_VALUE As String
                     search_TYPE As Short) As String
```

5.2.2 Parameters

dsn_NAME	the dsn to use to connect to database
table_NAME	the table within database to use
coumn_lNAME	the column to search
search_VALUE	the string to search for
searchTYPE	the type of search 0 – Begins with 1 – Contains

5.2.3 Returns

If a match found	the matching string
If no match found	""
If error	**Error description**

5.3 isEXVAL

5.3.1 Overview

This function is passed a set of parameters and does a search within a database/column. The purpose is to establish whether a value exists within that table/column, and no other data is returned.

```
Function isEXVAL(    dsn_NAME As String
                     table_NAME As String
                     search_COLUMN As String
                     search_VALUE As String) As Boolean
```

5.3.2 Parameters

dsn_NAME	the dsn to use to connect to database
table_NAME	the table within database to use
column_NAME	the column to search
search_VALUE	the string to search for

5.3.3 Returns

If a match found	True
If no match found	False

5.4 dbMATCHMULT

5.4.1 Overview

This function is passed a column to search, and a value to search for within the column. It can also be specified which columns from the table to return. Up to 3 columns can be returned. One of these can be the search column, although this is not necessary.

If desired, the value from any of the three returned columns can be displayed in the selection list shown where there are multiple matches.

The returned string will contain all three column values concatenated with the separator string which is also passed. The separator string is passed to ensure that no conflict will occur with the data held in the table.

Example: separator="%%", returned string could look like:

"1234%%Peter Smith%%Flat 21a, Manchester Rd"

```
Function dbMATCHMULT(       dsn_NAME As String
                            table_NAME As String
                            search_COLUMN As String
                            return_COLUMN1 As String
                            return_COLUMN2 As String
                            return_COLUMN3 As String
                            search_VALUE As String
                            separator_STRING As String
                            show_RETURN_COLUMN As Short) As String
```

5.4.2 Parameters

dsn_NAME	the dsn to use to connect to database
table_NAME	the table within database to use
search_COLUMN	the column to search
return_COLUMN1	)the columns to return. One can be the same as search
return_COLUMN2	)column if required
return_COLUMN3	)return_COLUMN1 is mandatory, return_COLUMN2/3 can be "" if not required
search_VALUE	the value to match
separator_STRING	a separator string for the returned parts
show_RETURN_COLUMN	flag to determine whether to show return_COLUMN2 value 0 – search_COLUMN value 1 – search_COLUMN value | return_COLUMN1 value 2 – search_COLUMN value | return_COLUMN2 value 3 – search_COLUMN value | return_COLUMN3 value

5.4.3 Returns

If a match found	the matching string
If no match found	""
If error	**Error description**

5.5 dbCORRECT

5.5.1 Overview

This function allows the searching within a database table and column for strings matching a search string with a certain confidence level.

This can be used as a flexible lookup facility, or as an auto-correction utility.

The search type can be first match, best match, or all matches. Using the latter allows the user to see a selection of all strings which exceeded the confidence level specified. This allows the lookup functionality.

Using either first match or best match returns a maximum of one result and can be used to auto-populate and correct either OCR results, or user entered data.

Example:

Search for **m4nchester** as typical OCR error

First/best match may return: MANCHESTER

All matches may return: MANCHESTER
WINCHESTER
DORCHESTER

Results will depend on the confidence level specified and the length of the search string passed. The closer in length to the required value, the higher the confidence level will be.

```
Function dbCORRECT(         dsn_NAME As String
                            table_NAME As String
                            column_NAME As String
                            search_VALUE As String
                            confidence_LEVEL As Double
                            search_TYPE As Short) As String
```

5.5.2 Parameters

dsn_NAME	the name of the system DSN used to connect to the database
table_NAME	the table to search in
column_NAME	the column to search
search_VALUE	the string to be matched
confidence_LEVEL	the confidence level to be matched
search_TYPE	the type of search 0 – first match (above confidence specified) 1 – best match (highest confidence) 2 – all matches (above specified confidence)

5.5.3 Returns

If a match found	the matching string
If no matches found	""
If error	**Error description**

5.6 txtCORRECT

5.6.1 Overview

This function allows the searching within a text file for strings matching a search string with a certain confidence level.

This can be used as a flexible lookup facility, or as an auto-correction utility.

The search type can be first match, best match, or all matches. Using the latter allows the user to see a selection of all strings which exceeded the confidence level specified. This allows the lookup functionality.

Using either first match or best match returns a maximum of one result and can be used to auto-populate and correct either OCR results, or user entered data.

Example:

Search for **m4nchester** as typical OCR error

First/best match may return: MANCHESTER

All matches may return: MANCHESTER
WINCHESTER

Results will depend on the confidence level specified and the length of the search string passed. The closer in length to the required value, the higher the confidence level will be.

```
Function txtCORRECT(        textfile_PATH As String
                            search_VALUE As String
                            confidence_LEVEL As Double
                            search_TYPE As Short) As String
```

5.6.2 Parameters

textfile_PATH	the text file to search
search_VALUE	the string to be matched
confidence_LEVEL	the confidence level to be matched
search_TYPE	the type of search 0 – first match (above confidence specified) 1 – best match (highest confidence) 2 – all matches (above specified confidence)

5.6.3 Returns

If a match found	the matching string
If no mathes found	“”
If error	**Error description**

5.7 txtLOOKUP

5.7.1 Overview

This function is passed a string does a search within a text file. A flag is set to determine whether the lookup uses a 'begins with' or 'contains' search.

Where there is a single match, the value is passed automatically. Where there are multiple matches, a selection list is displayed for a manual selection.

```
Function txtLOOKUP(  textfile_PATH As String
                     search_VALUE As String
                     search_TYPE As Short) As String
```

5.7.2 Parameters

textfile_PATH	the path to the file to search in
search_VALUE	the string to search for
search_TYPE	the type of search 0 – Begins with 1 – Contains

5.7.3 Returns

If a match found	the matching string
If no match found	""
If error	**Error description**

5.8 dbMATCH_Two_Fields

5.8.1 Overview

This function is used to match 2 values from a database table. It is possible to specify for each column whether to search EQUALS | NOT EQUALS | BEGINS | CONTAINS

It is then possible to specify whether the 2 criteria are used in an AND or OR search.

Up to 5 return columns can be specified, as well as a separator string.

```
Function dbMATCH_Two_Fields(      dsn_Name As String
                                  table_Name As String
                                  field1_Column As String
                                  field1_SearchValue As String
                                  field1_MatchType As Integer
                                  field2_Column As String
                                  field2_SearchValue As String
                                  field2_MatchType As String
                                  search_Type As Integer
                                  return_Column1 As String
                                  return_Column2 As String
                                  return_Column3 As String
                                  return_Column4 As String
                                  return_Column5 As String
                                  separator_String As String) As String
```

5.8.2 Parameters

dsn_Name	the dsn to use
table_Name	the table to search in
field1_Column	first search column
field1_SearchValue	first column search value
field1_MatchType	0 - equal 1 - not equal 2 - BEGINS 3 - CONTAINS
field2_Column	second search column
field2_SearchValue	second column search value
field2_MatchType	0 - equal 1 - not equal 2 - BEGINS 3 - CONTAINS
search_Type	0 - AND 1 - OR
return_Column1	return column 1
return_Column2	return column 2
return_Column3	return column 3
return_Column4	return column 4
return_Column5	return column 5
separator_String	return value separator

5.8.3 Returns

If a match found	the matching values with separators
If no match found	“”
If error	**Error description**

5.9 fileBrowser

5.9.1 Overview

This function is used to call a widows File Browser, and the selected file is returned to the calling code.

```
Function fileBrowser (      base_Path As String
                            browser_Title As String
                            return_Type As Integer) As String
```

5.9.2 Parameters

base_Path	the starting path for the browser
browser_Title	the title to display in the browser
return_Type	0 – the full path 1 – the filename

5.10 folderBrowser

5.10.1 Overview

This function is used to call a windows Folder Browser, and the selected folder is return to the calling code.

```
Function folderBrowser(     allow_Create As Boolean
                            browser_Title As String) As String
```

5.10.2 Parameters

allow_Create	TRUE – the user can create folders FALSE – the user cannot create folders
browser_Title	the title to display in the browser

5.11 spellChecker

5.11.1 Overview

This function is used to call a spell checking facility. A large volume of text can be passed, and if there are errors in the text, a display can be shown where a user will see underlined words as per normal windows spell checking.

The corrected text is passed back to the calling code.

```
Function spellChecker(      text_To_Check As String
                            custom_Dictionary As String
                            display_Title As String
                            always_Show As Boolean) As String
```

5.11.2 Parameters

text_To_Check	the text to be checked
custom_Dictionary	a custom dictionary file (see 6.5 below)
display_Title	the title to display in the spell checker
always_Show	TRUE – will be displayed even when no spelling errors FALSE – only show when there are spelling errors

5.12 flexSEARCH

5.12.1 Overview

This function is used to call a fully interactive database lookup utility where a user can perform a multi field 'exact' or 'contains' search in a database.

5.12.2 Parameters

```
Function flexSEARCH  (      dsn_Name As String
                            table_Name As String
                            column_Name_1-n As String *
                            displayName_1-n As String *
                            search_Type As Integer
                            separator_String As String
                            search_Title As String
                            ) As String
```

dsn_Name	the name of the dsn used to connect to the data source
table_Name	the table within the datasource
column_Name_1-n*	the column(s) to be included
display_Name_1-n*	the display name(s) for columns
search_Type	0 – AND 1 – OR
separator_String	the separator to be used in return string
search_Title	the form title for the interactive display

* The number of search columns is unlimited, but they must always be in pairs of "column name", "display name" so the following function calls are all valid from VB.Net.

```
dcUTIL_DotNet.flexSEARCH("flexSEARCH_MDB", "names", "Title", "Title", "FirstName",
"First Name", "Surname", "Surname", "Address", "Address Line 1", "Address2", "Address
Line 2", "Town", "Town", "Postcode", "Postcode", "Award", "Award", 0, "__", "Search For
Client – Full Details")
```

```
dcUTIL_DotNet.flexSEARCH("flexSEARCH_MDB", "names", "FullName", "Full Name", 0, "**",
"Search For Client – Full Name")

dcUTIL_DotNet.flexSEARCH("flexSEARCH_MDB", "names", "Title", "Title", "FirstName",
"First Name", "Surname", "Surname", 0, "|", "Search For Client – Title and Name")
```

5.12.3 Return

If a match found	the matching values with separators
If no match found	""
If error	**Error description**

5.12.4 Using flexSEARCH with KTM

When using flexSEARCH with KTM, the following method must be used to pass the parameters in an array to flexSEARCH:

```
'create an array of the correct size
Dim paraARR(8)

'add the relevant parameters
paraARR(0)="flexSEARCH_SQL"
paraARR(1)="Client_Contacts"
paraARR(2)="Contact_REFERENCE"
paraARR(3)="Contact Number"
paraARR(4)="Company_IDENTIFIER"
paraARR(5)="Company Name"
paraARR(6)=0
paraARR(7)="|"
paraARR(8)="Select a Vendor"

'pass the parameter array to flexSEARCH function
pField.Text=dcUTIL_DotNet.flexSEARCH(paraARR)
```

5.12.5 Using flexSEARCH with SBL

When using flexSEARCH with SBL, the following method must be used to pass the parameters. Note that the function to be used in SBL is called **flexSEARCH_SBL**

```
EnteredValue=dcUTIL_DotNet.flexSEARCH_SBL("dcUTIL_SQL", "Client_Contacts",
"Contact Number||Contact Number||Company Name||Company Name||Add Line
1||Address 1||Add Line 2||Address 2", 0, "__","Please Select Vendor")
```

The column names and display names are put together into a single string delimited by ||, and passed as a single value. You are still able to define as many columns as required.

6 General Considerations

6.1 Permissions

Permissions will be required from any workstation using any of the calls within dcUTIL_DotNet to any text file or database specified.

Permissions include the ability for a user/station to browse and read the files and/or databases, and may also require specific settings to be added to database security.

Please refer to your system administrator if any issues are encountered.

6.2 Text File Sizes

dcUTIL_DotNet can be used to search within text files as well as database tables. Where this facility is used, searches in large text files will be slower than those in databases of similar sizes.

Performance will vary between workstations, as a result of different performances of these. It is recommended that a variety of benchmark tests be carried out to ensure that the performance is acceptable.

Where testing shows performance to be lower than required, consider importing text data into a database table and using the corresponding database functionality instead.

6.3 Text File Format

The format of the text files used for searching is one entry per line. Any characters on a line will be considered part of the data.

6.4 Confidence Levels

Where correction/lookup functionality is used with a specified confidence level, the confidence level is calculated using a string difference algorithm. This algorithm calculates the number of changes required to match 2 strings. This returns a value known as the string difference. String difference can be considered as the number of additions/deletions/changes required to make string1 the same as string2.

Examples:

1. m4nche5ter and Manchester	string difference is 2 (4-a, 5-s)
2. 4nche5ter and Manchester	string difference is 3 (add m, 4-a, 5-s)

To calculate the confidence level, the following formula is used:

$$\frac{\text{Length of String (from text file or database)} - \text{String Difference}}{\text{Length of String (from text file or database)}}$$

This would give confidences of 80% and 70% for examples 1 and 2 above.

It is important to note that the greater the difference in length between two strings, the lower the confidence level will be. Where this may cause an issue with auto-correction, it is recommended to use the dbCORRECT or txtCORRECT calls with the search type set to ALL, and allow a user to select the correct value from a list.

It is possible to mix several types of search in order to get the best results, and the most automated functionality. For example:

At Recognition use auto-correction with a high confidence level and BEST as search type. If no matches are found, keep the original OCR results.

Then during Validation, use the same function but with lower confidence and the search type set to ALL. This will allow more chance of finding the value, and allow a user to verify from a list if multiple matches are found.

6.5 Custom Dictionaries

When calling the Spell Checker function, if the **custom_Dictionary** paramenter is left as "", then the standard windows dictionary will be used.

In some cases you may need to specify a custom dictionary to allow specific words required for your application.

Custom dictionaries use lexicon files, which are text files that have a .lex extension. Each line of a lexicon file contains a single word that is accepted as a legitimate spelling. The first line of the file can specify a locale identifier (LCID) that the dictionary applies to. If the locale is not specified, the dictionary applies to all languages.

Custom dictionaries are used in addition to the default spelling checker

7 Contact

For help and support contact **mail@davidcrewe.com**