dcUTIL_KTM

Script utility collection for use with Kofax Transformation Modules

dcUTIL_KTM Supports KTM 5.5 and above

Contents

1 Overview

dcUTIL_KTM is a collection of script utilities developed for use with Kofax Transformation Modules (KTM) scripts.

These script utilities allow complex batch routing, document ordering, document merging, and document deleting processes to be added to KTM projects with little scripting knowledge.

A script reference is made to the dcUTIL_KTM plug-in, and then you can call any of the functions easily from within your KTM scripts.

Processes which would normally take hours or days to develop can be added in minutes.

2 Function Calls within dcUTIL_KTM

2.1 SplitBatchByClass

Allows a batch to be split into multiple new batches – each corresponding to a different document class. Documents from the original batch are moved to relevant new batches.

2.2 SplitBatchByConfidence

Allows a batch to be split into a Low Confidence batch and a High Confidence batch. A threshold is specified and documents are moved into the relevant new batch.

Custom routing can be applied for each of the new batches.

2.3 SplitBatchByIndexFieldValue

Allows you to split documents with a specific index field value into a new batch. All documents with the specified value will be moved to the new batch.

Custom routing can be applied to the new batch.

2.4 SplitBatchByNumberOfDocuments

Allows a batch to be split into smaller batches. You can specify the maximum number of documents to be moved to each new batch.

2.5 SplitBatchByDistinctIndexValue

Allows a batch to be split into multiple new batches, each relating to a distinct index field value.

You specify the index field to check, and a new batch is created for each distinct value. All documents sharing the same value will be moved to the same new batch.

2.6 SplitBatchByDate

Allows a batch to be split into two new batches, based on whether an index field has a date before or after a check date.

You specify the date to check, and the index field holding a date, and new batches are created for document with a date before or after the check date.

2.7 DeleteDocumentsByClass

Allows documents of specified class(es) to be removed from a batch.

2.8 DeleteDocumentsByIndexFieldValue

Allows documents with specific index field values to be removed from a batch.

2.9 DeleteDocumentsByDateBefore

Allows documents to be deleted if the date in a specified index field is before a cut-off date.

2.10 DeleteDocumentsByDateAfter

Allows documents to be deleted if the date in a specified index field is after a cut-off date.

2.11 MergeDocumentsByClass

Allows documents of the same class to be merged into a single document.

2.12 MergeDocumentsByIndexFieldValue

Allows documents which share a common index field value to be merged.

2.13 OrderDocumentsByClass

Allows documents to be reordered within a batch, so that all documents of the same class are arranged together.

2.14 OrderDocumentsByDateAscending

Allows documents to be reordered within a batch, in ascending date order, based on dates found in a specified index field.

2.15 OrderDocumentsByDateDescending

Allows documents to be reordered within a batch, in descending date order, based on dates found in a specified index field.

2.16 OrderDocumentsByIndexFieldValue

Allows documents to be reordered within a batch so that all documents sharing a common specified index field value are arranged together.

2.17 OrderDocumentsByIndexFieldAscending

Allows documents to be reordered into ascending alphabetical order within a batch, based on values in a specified index field.

2.18 OrderDocumentsByIndexFieldDescending

Allows documents to be reordered into descending alphabetical order within a batch, based on values in a specified index field.

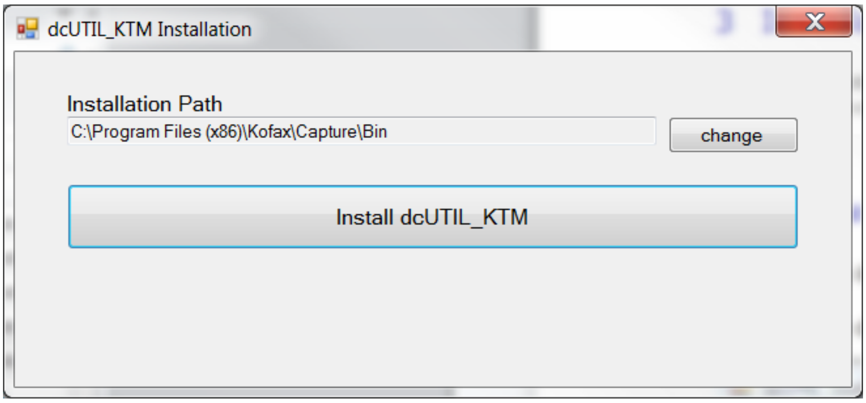
3 Installation and Implementation

In order to use the utilities within dcUTIL_KTM, the dll must be installed and registered on each machine where the calling script will be run.

3.1 Install dcUTIL_KTM

The steps to install dcUTIL_KTM are as follows:

1. Unzip the dcUTIL_KTM.zip folder, and run the **dcUTIL_KTM_Installer.exe**



C:\Program Files (x86)\Kofax\Capture\Bin

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

3. Install dcUTIL_KTM

4. Repeat on each workstation which will require dcUTIL_KTM functionality.

You may require Administrative privileges to register dcUTIL_KTM. Please check with your IT Administrator if you experience any problems registering.

3.2 Logging

By default dcUTIL_KTM will log errors encountered to an error log file located in the dcUTIL_KTM_LOGS folder in the installation directory.

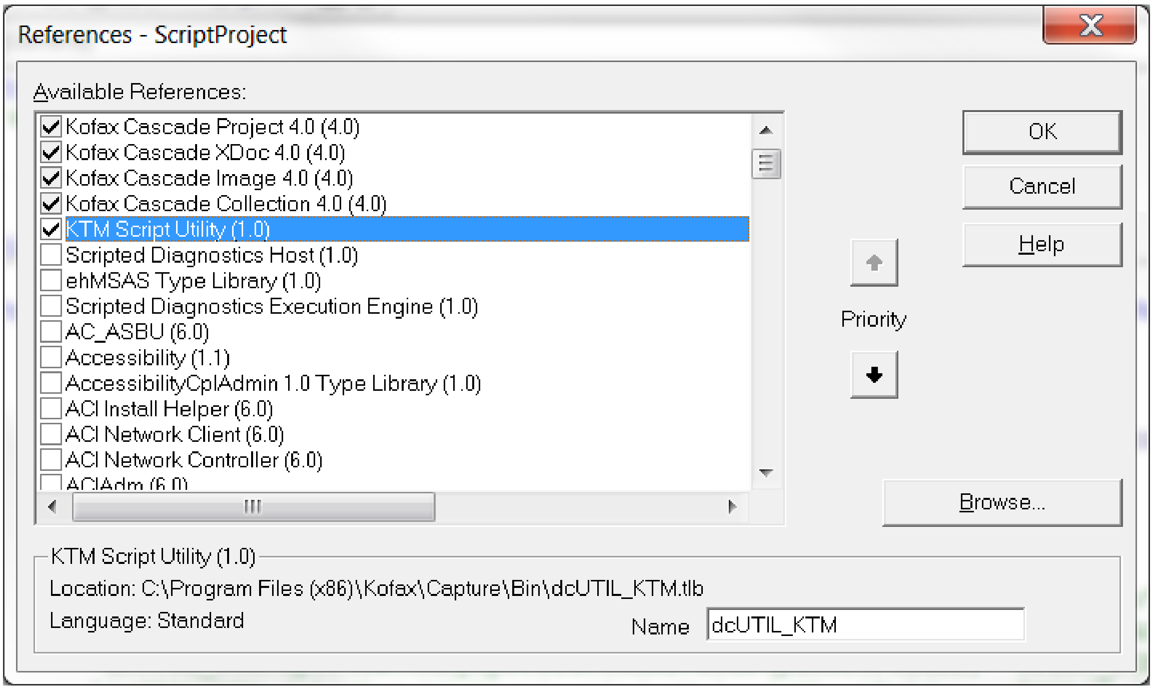
4 Incorporate into KTM Script

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

Open the Project level script, and select Edit > References

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

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



References - ScriptProject

Available References:

☑ Kofax Cascade Project 4.0 (4.0)
☑ Kofax Cascade XDoc 4.0 (4.0)
☑ Kofax Cascade Image 4.0 (4.0)
☑ Kofax Cascade Collection 4.0 (4.0)
☑ KTM Script Utility (1.0)
□ Scripted Diagnostics Host (1.0)
□ ehMSAS Type Library (1.0)
□ Scripted Diagnostics Execution Engine (1.0)
□ AC_ASBU (6.0)
□ Accessibility (1.1)
□ AccessibilityCplAdmin 1.0 Type Library (1.0)
□ ACI Install Helper (6.0)
□ ACI Network Client (6.0)
□ ACI Network Controller (6.0)
□ ACIAdm (6.0)

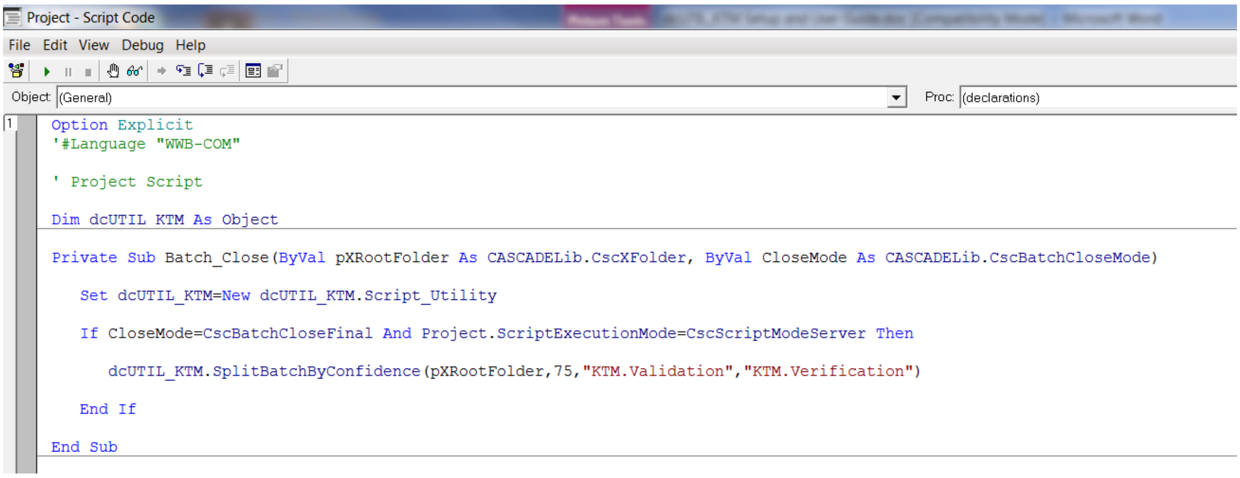
KTM Script Utility (1.0)

Location: C:\Program Files (x86)\Kofax\Capture\Bin\dcUTIL_KTM.tlb

Language: Standard

Name: dcUTIL_KTM

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



```
Option Explicit
'#Language "WWB-COM"

' Project Script

Dim dcUTIL_KTM As Object

Private Sub Batch_Close(ByVal pXRootFolder As CASCADELib.CscXFolder, ByVal CloseMode As CASCADELib.CscBatchCloseMode)

   Set dcUTIL_KTM=New dcUTIL_KTM.Script_Utility

   If CloseMode=CscBatchCloseFinal And Project.ScriptExecutionMode=CscScriptModeServer Then

      dcUTIL_KTM.SplitBatchByConfidence(pXRootFolder,75,"KTM.Validation","KTM.Verification")

   End If

End Sub
```

In the example above the following relevant code is used:

1. Add ***'#Language "WWB-COM"*** at the top of the script.

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

Dim dcUTIL_KTM As Object

3. Add code to the relevant project level event to create an instance of dcUTIL_KTM:

Set dcUTIL_KTM=New dcUTIL_KTM.Script_Utility

4. Call the relevant functions from the created object.

In the example above we call the ***SplitBatchByConfidence*** function on Batch_Close in KTM Server and pass parameters for the confidence level, and the routing for low and high confident batches.

Any documents with an average confidence less than 75% will be moved to a separate batch and routed to KTM Validation.

Any documents with an average confidence 75% or above will be moved to a separate batch and routed to KTM Verification.

5 Function Call Details

5.1 Overview

5.1.1 pXRootFolder

The calls in dcUTIL_KTM all make use of the pXRootFolder object which is always passed as a parameter to each function call.

The object pXRootFolder in KTM roughly equates to the Batch in Kofax Capture, so when we pass this object we are able to make changes to the batch structure, which are then passed back to KTM from dcUTIL_KTM with any appropriate changes made.

dcUTIL_KTM calls can be made from any KTM script event which exposes the pXRootFolder object.

In most cases the functionality would be called from Batch_Open or Batch_Close at the Project level.

5.1.2 Batch Object

All calls connected with merging documents require the 'Batch' parameter to be passed. This is a script object allowing certain batch operations to be performed.

5.1.3 Splitting Batches

Where batches are split in dcUTIL_KTM there may be the option to specify a specific name for the newly created batches. For example with an original batch called 'TestDocs', which has the SplitBatchByClass call applied, we could end up with new batches with the class as a suffix. For example:

-TestDocs Northwest
-TestDocs TriSpectrum

Where the naming cannot be explicitly specified, the newly created batches will be numbered, so for a batch called ‘TestDocs’ you may end up with newly created batches such as:

-TestDocs 1
-TestDocs 2

5.1.4 Routing Batches

Some function calls allow you to specify the routing for newly created batches. There are a set of values which must be used when using this feature, and these are shown below:

KTM.Server
KTM.Server2
KTM.DocumentReview
KTM.Correction
KTM.Validation
KTM.Validation2
KTM.Validation3
KTM.KBLearningServer
KTM.Verification
KC.Scan
KC.Export
KC.Recognition
KC.Validation
KC.PDF
KC.OCR
KC.QC
KC.Verification

5.1.5 CloseMode and ScriptExecutionMode

When calling dcUTIL_KTM functionality, we need to be able to determine where and when during the KTM workflow it is called.

The script snippet below shows 2 criteria we can check.

```
Private Sub Batch_Close(ByVal pXRootFolder As CASCADELib.CscXFolder, ByVal CloseMode As
CASCADELib.CscBatchCloseMode)

  Set dcUTIL_KTM=New dcUTIL_KTM.Script_Utility

  If CloseMode=CscBatchCloseFinal And Project.ScriptExecutionMode=CscScriptModeServer Then

    dcUTIL_KTM.SplitBatchByConfidence(pXRootFolder,75,"KTM.Validation"," KC.PDF")

  End If

End Sub
```

CloseMode is the way in which the batch is closing (suspend, final etc). In most cases we need to call dcUTIL_KTM on final close, so use the test as shown above.

ScriptExecutionMode lets us check where the script is running, and takes one of the following values.

CscScriptModeCorrection	The script is executed by the Correction runtime.
CscScriptModeDocumentReview	The script is executed by the Document Review runtime.
CscScriptModeServer	The script is executed by the server.
CscScriptModeServerDesign	The design application executes the server part.
CscScriptModeUnknown	This value was not initialized.
CscScriptModeValidation	The script is executed by the validation runtime.
CscScriptModeValidationDesign	The design application executes the validation part.
CscScriptModeVerification	The script is executed by the Verification runtime.
CscScriptModeVerificationDesign	The script is executed by the Verification designer.

5.2 SplitBatchByClass

This call will split out any documents where the KTM extraction class is in the strClassList parameter. Where "" is passed in this parameter, ALL classes will be split into new batches.

After running, there will be a new batch for each class specified, and all documents of that class will be moved to the relevant new batch.

```
SplitBatchByClass     (pXRootFolder,
                       strClassList,
                       intNameType)
```

5.2.1 Parameters

pXRootFolder	the root folder object
strClassList	the classes to split eg. 'Northwest|TriSpectrum|OtherClass'
intNameType	the naming for new batches 1 – class name as suffix 0 - numeric

5.2.2 Example Call

dcUTIL_KTM.SplitBatchByClass(pXRootFolder,"Northwest|TriSpectrum",1)

Any documents of either of the two specified classes will be split out into new batches – one new batch for each class.

5.3 SplitBatchByConfidence

This call will split a batch according to a specified confidence level. All documents below the threshold will be moved into the 'low' batch, and all those above the threshold will be moved into the 'high' batch.

Custom routing may be specified for each new batch. If none is specified, the new batches will follow the standard workflow.

```
SplitBatchByConfidence      (pXRootFolder,
                            intConfidence,
                            strLowRouting,
                            strHighRouting)
```

5.3.1 Parameters

pXRootFolder	the root folder object
intConfidence	the confidence threshold
strLowRouting	next module for low confidence batch
strHighRouting	next module for high confidence batch

5.3.2 Example Call

dcUTIL_KTM.SpitBatchByConfidence(pXRootFolder,70,"KTM.Validation","KC.PDF")

in the example above, all documents with a confidence of less than 70% will be moved to a new batch and routed to KTM validation. All documents with a higher confidence will be moved to another batch and routed to PDF Generator.

If the strLowRouting or strHighRouting parameters are passed as "" then the standard workflow will be applied.

5.4 SplitBatchByDistinctIndexValue

This call will create a new batch for each distinct value found in the index field specified. Any documents which do not have that index field will be ignored.

```
SplitBatchByDistinctIndexValue      (pXRootFolder,
                                    strFieldName)
```

5.4.1 Parameters

pXRootFolder	the root folder object
strFieldName	the index field to use

5.4.2 Example Call

dcUTIL_KTM.SplitBatchByDistinctIndexValue(pXRootFolder,”Split_Value”)

In the example above, a new batch is created for each distinct Split_Value index field value in the original batch. All documents with the same value will be moved to the same new batch.

5.5 SplitBatchByDate

This call will create two new batches, based on whether a specified index field has a date before or after a check date. Any documents without the index field specified will be ignored.

```
SplitBatchByDate              (pXRootFolder,
                               strDateValue,
                               strDateField,
                               strPreRouting,
                               strPostRouting)
```

5.5.1 Parameters

pXRootFolder	the root folder object
strDateValue	the check date
strDateField	the index field holding the document level date
strPreRouting	routing module for the Pre date batch
strPostRouting	routing module for the Post date batch

5.5.2 Example Call

dcUTIL_KTM.SplitBatchByDate(pXRootFolder,”01 January 2018”,”DocDate”,””,”KC.PDF”)

In the example above, new batches could be created for documents with dates before or after the check date – 01 January 2018. For an original batch called TestDocs you could end up with:

-TestDocs Pre 01 January 2018	uses standard workflow as next module
-TestDocs Post 01 January 2018	routed to PDF Generator

Leaving the routing parameters as “” will mean the new batches follow the standard workflow.

Where the date is the same as the check date, the document will be moved to the Post batch.

5.6 SplitBatchByNumberOfDocuments

This function splits a batch into new batches of a specified size.

```
SplitBatchByNumberOfDocuments  (pXRootFolder,
                                intSplitNumber)
```

5.6.1 Parameters

pXRootFolder	the root folder object
intSplitNumber	the number of documents per batch

5.6.2 Example Call

```
dcUTIL_KTM.SplitBatchByNumberOfDocuments (pXRootFolder,10)
```

In the example above, if you scanned an original batch of 25 documents called TestDocs, you would end up with the following batches:

-TestDocs	10 documents
-TestDocs 1	10 documents
-TestDocs 2	5 documents

5.7 SplitBatchByIndexFieldValue

This function splits a batch based on a specific index field value. All documents with the specified index field value will be moved to a new batch.

```
SplitBatchByIndexFieldValue     (pXRootFolder,
                                 strFieldName,
                                 strFieldValue,
                                 intNameType,
                                 strRouting)
```

5.7.1 Parameters

pXRootFolder	the root folder object
strFieldName	the index field to check
strFieldValue	the value to match
intNameType	0 – numeric suffix for new batch 1 – index value suffix for new batch
strRouting	custom routing module

5.7.2 Example Call

dcUTIL_KTM.SplitBatchByIndexFieldValue (pXRootFolder,"SplitDoc","YES",1,"KC.PDF")

In the example above all documents where the index field SplitDoc has the value YES, will be moved to a new batch. If the original batch was called TestDocs, then new batch would be called 'TestDocs YES'.

5.8 DeleteDocumentsByClass

This function allows you to remove any documents of the specified class(es)

```
DeleteDocumentsByClass    (pXRootFolder,
                           strClassList)
```

5.8.1 Parameters

pXRootFolder	the root folder object
strClassList	the class(es) to remove

5.8.2 Example Call

dcUTIL_KTM.DeleteDocumentsByClass (pXRootFolder,"Northwest|TriSpectrum")

In the example above, all documents with an extraction class of Northwest or TriSpectrum will be removed from the batch.

5.9 DeleteDocumentsByIndexFieldValue

This function allows you to remove any documents which have a specific index field value.

```
DeleteDocumentsByIndexFieldValue    (pXRootFolder,
                                     strFieldName,
                                     strFieldValue)
```

5.9.1 Parameters

pXRootFolder	the root folder object
strFieldName	the index field to check
strFieldValue	the value to check for

5.9.2 Example Call

dcUTIL_KTM.DeleteDocumentsByIndexFieldValue(pXRootFolder,"Remove","YES")

In the example above, any documents which have an index field called Remove, which has a value of YES will be removed from the batch.

5.10 DeleteDocumentsByDateBefore

This function allows you to remove any documents which where the date in a specified index field is earlier than a specified cut-off date.

```
DeleteDocumentsByDateBefore      (pXRootFolder,
                                  strDateValue,
                                  strDateField)
```

5.10.1 Parameters

pXRootFolder	the root folder object
strDateValue	the cut-off date
strDateField	the index field to check for date

5.10.2 Example Call

dcUTIL_KTM.DeleteDocumentsByDateBefore(pXRootFolder,"01 January 2018","RecdDate")

In the example above, any documents where the date in index field RecdDate is earlier than 01 January 2018 will be removed.

5.11 DeleteDocumentsByDateAfter

This function allows you to remove any documents which where the date in a specified index field is later than a specified cut-off date.

```
DeleteDocumentsByDateAfter       (pXRootFolder,
                                  strDateValue,
                                  strDateField)
```

5.11.1 Parameters

pXRootFolder	the root folder object
strDateValue	the cut-off date
strDateField	the index field to check for date

5.11.2 Example Call

dcUTIL_KTM.DeleteDocumentsByDateBefore(pXRootFolder,"01 January 2018","RecdDate")

In the example above, any documents where the date in index field RecdDate is later than 01 January 2018 will be removed.

5.12 MergeDocumentsByClass

This function allows you to merge documents of the same class into a single document.

```
MergeDocumentsByClass          (pXRootFolder,
                                Batch,
                                strClassList)
```

5.12.1 Parameters

pXRootFolder	the root folder object
Batch	the script batch object
strClassList	a | delimited list of classes which should be merged

5.12.2 Example Call

dcUTIL_KTM. MergeDocumentsByClass (pXRootFolder,Batch,"Northwest|TriSpectrum")

In the example above, all documents of Northwest will be merged into a single document, and all documents of TriSpectrum will be merged into a single document.

If the original batch had documents:

Northwest	3 pages
TriSpectrum	2 pages
TriSpectrum	2 pages
Northwest	3 pages
Northwest	3 pages

The resulting batch would have:

Northwest	9 pages
TriSpectrum	4 pages

The order of pages would be maintained.

If the strClassList parameter is left as "" then all classes will be merged.

5.13 MergeDocumentsByIndexFieldValue

This function allows you to merge documents which share a common index field value into single documents.

```
MergeDocumentsByIndexFieldValue        (pXRootFolder,
                                        Batch,
                                        strFieldName)
```

5.13.1 Parameters

pXRootFolder	the root folder object
Batch	the script batch object
strFieldName	the field to check for common values

5.13.2 Example Call

```
dcUTIL_KTM. MergeDocumentsByIndexFieldValue(pXRootFolder,Batch,"Merge_Field")
```

In the example above, all documents of any class which have the same value in the Merge_Field index field will be merged into a single document.

If the original batch had documents:

Northwest	3 pages	Merge_Field=Y
TriSpectrum	2 pages	Merge_Field=Y
TriSpectrum	2 pages	Merge_Field=N
Northwest	3 pages	Merge_Field=N
Northwest	3 pages	Merge_Field=Y

The resulting batch would have:

Northwest	8 pages
TriSpectrum	5 pages

Where documents are merged, the classification of the first document is maintained.

The order of pages would be maintained.

5.14 OrderDocumentsByClass

This function allows you to reorder the documents with a batch according to the extraction class(es) specified.

```
OrderDocumentsByClass     (pXRootFolder,
                           strClassList)
```

5.14.1 Parameters

pXRootFolder	the root folder object
strClassList	the classes to reorder

5.14.2 Example Call

```
dcUTIL_KTM.OrderDocumentsByClass(pXRootFolder,”Northwest|TriSpectrum”)
```

In the example above, the batch will be reordered so that all Northwest documents are together, and all TriSpectrum documents are together. All other document classes will be ignored and stay in their original positions.

If you leave the strClassList as “”, then ALL classes will be reordered.

5.15 OrderDocumentsByDateAscending

This function allows you to reorder the documents with a batch according to dates found in a specified field.

```
OrderDocumentsByDate       (pXRootFolder,
                           strFieldname)
```

5.15.1 Parameters

pXRootFolder	the root folder object
strFieldname	the index field to check for dates

5.15.2 Example Call

```
dcUTIL_KTM.OrderDocumentsByDate(pXRootFolder,”Due_Date”)
```

In the example above, the documents in the batch will be reordered in ascending date order, based on the value in the Due_Date field.

Documents without this field will be ignored.

5.16 OrderDocumentsByDateDescending

This function allows you to reorder the documents with a batch according to dates found in a specified field.

```
OrderDocumentsByDate       (pXRootFolder,
                           strFieldname)
```

5.16.1 Parameters

pXRootFolder	the root folder object
strFieldname	the index field to check for dates

5.16.2 Example Call

dcUTIL_KTM.OrderDocumentsByDate(pXRootFolder,”Due_Date”)

In the example above, the documents in the batch will be reordered in descending date order, based on the value in the Due_Date field.

Documents without this field will be ignored.

5.17 OrderDocumentsByIndexFieldValue

This function allows you to reorder documents within a batch based on shared index field values.

```
OrderDocumentsByIndexFieldValue (pXRootFolder,
                                 strFieldName)
```

5.17.1 Parameters

pXRootFolder	the root folder object
strFieldName	the index field to evaluate

5.17.2 Example Call

dcUTIL_KTM.OrderDocumentsByIndexFieldValue (pXRootFolder,”ClientReference”)

In the example above, the batch will be reordered so that all documents which have the same value in the ClientReference field will be arranged together.

Any documents which do not have the ClientReference index field will not be moved.

5.18 OrderDocumentsByIndexFieldAscending

This function allows you to reorder documents alphabetically within a batch based on index field values.

```
OrderDocumentsByIndexFieldValue (pXRootFolder,
                                 strFieldName)
```

5.18.1 Parameters

pXRootFolder	the root folder object
strFieldName	the index field to evaluate

5.18.2 Example Call

dcUTIL_KTM.OrderDocumentsByIndexFieldAscending (pXRootFolder,”Client_Surname”)

In the example above, the batch will be sorted into in ascending alphabetical order based on the values in the Client_Surname field.

Any documents which do not have the Client_Surname index field will not be moved.

5.19 OrderDocumentsByIndexFieldDescending

This function allows you to reorder documents alphabetically within a batch based on index field values.

```
OrderDocumentsByIndexFieldValue (pXRootFolder,
                                 strFieldName)
```

5.19.1 Parameters

pXRootFolder	the root folder object
strFieldName	the index field to evaluate

5.19.2 Example Call

dcUTIL_KTM.OrderDocumentsByIndexFieldAscending (pXRootFolder,”Client_Surname”)

In the example above, the batch will be sorted into in descending alphabetical order based on the values in the Client_Surname field.

Any documents which do not have the Client_Surname index field will not be moved.

6 General Considerations

6.1 Confidence Levels

Where documents are split by confidence level, the confidence used is the average of all fields for the document. If there are empty fields this may have an effect on the overall confidence.

6.2 Multiple Runs

If a batch is processed multiple times through the same module – thereby triggering the same script to run multiple times – you may get unpredictable results. Ensure you fully understand the functionality and expected results, and test thoroughly in a development environment before use in production.

6.3 Merging Documents

When documents are merged, the classification of the first document is kept, and the classification of documents merged into it are ignored.

The order of pages is maintained.

6.4 Multiple Calls

It is possible to use more than one call from dcUTIL_KTM on the same batch, but thorough testing should be carried out to ensure that the desired results are obtained. In some cases conflicting settings could cause unexpected results.

An example where more than one call could be made would be:

```
dcUTIL_KTM.OrderDocumentsByDate(pXRootFolder,”Due_Date”)

dcUTIL_KTM. MergeDocumentsByClass (pXRootFolder,Batch,”Northwest|TriSpectrum”)
```

In this case the batch would first be sorted in order of Due_Date, and then documents of the same class would be merged. In this way, we could ensure that the pages within our merged documents are in date order.

7 Contact

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