Docket No. RM01-5-000
Electronic Tariff Filings

Word-To-RTF Extraction Application

The Supplemental NOPR in Docket No. RM01-5-000 proposes that certain FERC regulated tariffs must be converted to an electronic format and filed with the Commission. This document outlines a Microsoft Word to RTF extraction application (macro) that may be used to break up a large tariff document into smaller levels of granularity. The macro may be copied and adapted to different tariff document structures.

The use of a macro to perform the conversion task is not required. Indeed, tariff documents up to 300 to 400 pages converted to a granularity level of 1.0 (as the term is used in the Supplemental NOPR) may be done most efficiently by hand.

For conversion tasks that involve large tariff documents, a large number of tariff documents or a granularity level of 1.1 or greater, the use of a macro may be advisable. The example extraction macro shown below can extract tariff section text and save the tariff text as individual RTF files. The macro also creates a database which stores information about each section.

Macro Overview

The objective of the extraction macro is to speed-up the sectioning and conversion of tariffs to the require FERC tariff record filing format, RTF. This extraction macro assumes:

a) The tariff is in Word 2000 or greater
b) The tariff is formatted in a similar fashion throughout the entire document: section titles are bold and section numbers exist using either raw numbers or an automated numbered list. To find out more on numbered lists: search *Numbered Lists* using the help in Word.

Database File Specification

The extraction macro creates a database table with the following fields:

Field name	Data Type	Description
assoc_file	C(80)	The name of the associated RTF file (aka, the file containing a single section)
sect_id	N(6)	A unique number assigned by the program, used to uniquely identify a section
parent_id	N(6)	The sect_id (above) which the current section belongs to. For 'top level' sections this number should be 0
sect_num	C(50)	The tariff section number. This is not to be confused with the section ID. This number may be of the x.x.xx.xxx format.
sect_title	C(200)	The title of the tariff section. Do not include the tariff section 'number' as part of the title.

The table contains 1 record (row) for each RTF file. Each row in the table holds information regarding a single section of the tariff. Note that the Field Names used and Data Type requirements shown above are not the same as those provided in the proposed *Instruction Manual for Electronic Filing of Parts 35, 154, 284, 300, and 341 Tariff Filings.* Additional data elements and data manipulation will be required to complete the Tariff Record Content Data required for each Tariff Record.

Extraction Macro

This macro is intended to be a guideline or example of how a company's tariff document can be converted into the FERC's required format, RTF. Each company's tariff is maintained using various applications and will have a unique organizational format. For this reason customization of the extraction macro will be required for each document's source software and each document's organizational format.

Neither FERC nor its staff will provide, write, support or maintain extraction macros. There are too many variances of software, file formats and document structure to provide a standard macro applicable under all circumstances. The macro provided below is simply an example of what is possible.

Example Extraction Macro Detailed Design

This extraction macro performs three functions:

(1) Parse a tariff by section
(2) create an RTF file of each section
(3) Insert the relevant section information into a database table.

The extraction macro examines each paragraph in the document by looking at the characters at the beginning of the first line of each paragraph. There are two code listings at the end of this document, one is for paragraphs with manual numbering and the latter is for paragraphs with automated numbered lists. We recommend before attempting to run the any extraction macro, first standardize all formatting within your tariff. (i.e., bold all section headings, ensure all numbering is the same)

Macro VBA Code

The extraction macro VBA Code for parsing a Word-formatted tariff with manual numbering into the RTF format follows:

```
Option Explicit
'Declare the connection object
'NOTE:  Before you can use the connection and recordset objects
'You must set a Reference to the ADODB object library
Dim myconn As ADODB.Connection
Private Sub SearchThru()
'This sub cycles through all the paragraphs in the document,
'picking out the section numbers and section titles, etc.

Dim mypara As Paragraph
Dim mynewfile As Word.Document, myfile As Word.Document
Dim connstr As String, mystr As String
Dim i As Integer, j As Integer
Dim sectnum As String, assocfile As String, parent_sec As String
Dim sect_id As Integer, parent_id As Integer
Dim paratext As String, prevfile As String, secttitle As String, newtext As String
Dim mypath As String

On Error GoTo ErrHa

'Set the path
'Everything should reside in this directory
mypath = "D:\Test\"

'Build the connection string
connstr = "Provider=vfpoledb;Data Source=D:\Test\Word2eTariff.dbf;"

Set myconn = New ADODB.Connection 'set the object
myconn.Open connstr 'open the connection

'Set id vars
parent_id = 0
sect_id = 1

For Each mypara In ThisDocument.Paragraphs 'loop thru paragraph collection

  sectnum = "" 'reset vars for each iteration
  assocfile = ""

  'Clean the string
  If mypara.Range.Words.Count > 2 Then
    paratext = CleanString(mypara.Range.Text)
  Else
    'if the string isn't more than 2 chars then move on
```

```
    paratext = mypara.Range.Text
  End If

  'Sift thru the paragraph text 1 char at a time
  'searching for a numeric value or a "."
  For i = 1 To Len(paratext)
    If IsNumeric(Mid(paratext, i, 1)) Or Mid(paratext, i, 1) = "." Then

    Else 'no more dots or numbers
      'for paragraph without a section #
      'belongs to previous paragraph
      If i = 1 Then 'no section number, append to previous
        Set myfile = Documents.Open(mypath & prevfile) 'open the previous document
        myfile.Range.Text = myfile.Range.Text & paratext 'put text from existing document
        myfile.Save 'save the changes
        myfile.Close 'close the file
        Exit For 'break out of the For loop since our work is done here
      Else ' section number exists let's write to db
        sectnum = Left(paratext, i - 1) 'assign the section number

        'just in case it doesn't end in a "."
        If Right(sectnum, 1) <> "." Then
          sectnum = sectnum & "."
        End If

        'Find parent section
        If Len(sectnum) > 2 Then
          For j = Len(sectnum) - 1 To 1 Step -1
            If Mid$(sectnum, j, 1) = "." Then
              'found second "."
              parent_sec = Left(sectnum, j)
              'Look up the parent id
              parent_id = LookupParent(parent_sec)
              Exit For
            End If
          Next j
        Else
          'must be a top level
          parent_id = 0
        End If

        'Get title if it is present
        'All titles are bold in our test tariff

        If mypara.Range.Bold = True Or mypara.Range.Bold = "9999999" Then
          'secttitle = Right(mypara.Range.Text, Len(mypara.Range.Text) - i)
          secttitle = CleanString(Right(mypara.Range.Text, Len(mypara.Range.Text) - i))
        End If

        Set mynewfile = Documents.Add 'Create new document
        mynewfile.Range.Text = paratext 'Put text in new document

    mynewfile.SaveAs FileName:=mypath & sect_id & ".RTF", FileFormat:=wdFormatRTF 'save file

        mynewfile.Close 'close file
        prevfile = sect_id & ".RTF"

        assocfile = sect_id & ".RTF" 'set filename of doc based on the section number

        'Write into database
        mystr = "INSERT INTO Word2eTariff (assoc_file, sect_id, parent_id, sect_num, sect_title) VALUES ('" & assocfile & "', " & sect_id & ", "
& parent_id & ", '" & sectnum & "', '" & secttitle & "')"
        'If sect_id = 66 Then
        '    MsgBox mystr
        'End If

        myconn.Execute mystr

        sect_id = sect_id + 1 'Increment section ID
```

```
      Exit For
    End If

  End If

 Next i

Next mypara

End

ErrHa:
MsgBox mystr
End


End Sub
Private Function LookupParent(parent_sec As String) As Integer
'This function looks up the section id of a parent section
'The return value of this function will be used for the parent_id field of a new row

Dim myrst As ADODB.Recordset 'create a recordset to store rows for look up
'execute the SELECT statement, all rows will be returned in the recorset object
Set myrst = myconn.Execute("SELECT sect_id, sect_num FROM Word2eTariff")

'Loop thru the recordset and match parent_sec with sect_num
'and return the sect_id when found.
Do While Not myrst.EOF
 If Trim(myrst!sect_num) = Trim(parent_sec) Then
  LookupParent = myrst!sect_id 'return the sect_id
  Set myrst = Nothing
  Exit Function
 End If
 myrst.MoveNext 'move to the next record
Loop

MsgBox "Error!  Could not find section id to reference the parent id."

Set myrst = Nothing 'destroy objects = a good practice
LookupParent = 0 'return 0 if an error occurs; 0 = a top level section with no parent

End Function
Private Function CleanString(mystr As String) As String

'This function strikes all the chars that are non alpha and non numeric
'from the left and the right of the string that is the current paragraph.
'It returns a clean string.

'Dim i As Integer 'char counter
Dim done As Boolean 'flag variable

mystr = Trim(mystr) 'Trim spaces from left and right

'Clean Left side of the string
Do Until Asc(Left(mystr, 1)) > 33 And Asc(Left(mystr, 1)) < 126
 mystr = Right(mystr, Len(mystr) - 1)
Loop

'Clean Right side of the string
Do Until Asc(Right(mystr, 1)) > 33 And Asc(Right(mystr, 1)) < 126
 mystr = Left(mystr, Len(mystr) - 1)
Loop


'Return the cleaned string
CleanString = mystr


End Function
```

The extraction macro VBA Code for Parsing a Word-formatted with Numbered Lists tariff into the RTF format follows:

```
Option Explicit
'Declare the connection object
'NOTE:  Before you can use the connection and recordset objects
'You must set a Reference to the ADODB object library
Dim myconn As ADODB.Connection

Private Sub SearchThru()
'This sub cycles through all the paragraphs in the document,
'picking out the section numbers and section titles, etc.

Dim mypara As Paragraph
Dim mynewfile As Word.Document, myfile As Word.Document
Dim connstr As String, mystr As String
Dim i As Integer, j As Integer, dots As Integer
Dim sectnum As String, assocfile As String, parent_sec As String
Dim sect_id As Integer, parent_id As Integer
Dim prevfile As String, secttitle As String, newtext As String
Dim mypath As String

'Set the path
'Everything should reside in this directory
mypath = "D:\Test\"

'Build the connection string
connstr = "Provider=vfpoledb;Data Source=D:\Test\Word2eTariff.dbf;"

Set myconn = New ADODB.Connection 'set the object
myconn.Open connstr 'open the connection

'Set id vars
parent_id = 0
sect_id = 1


For Each mypara In ThisDocument.Paragraphs 'loop thru paragraphs collection
  sectnum = "" 'reset vars for each iteration
  assocfile = ""

  'Test for ListParagraph
  If mypara.Range.ListFormat.ListType > 0 Then 'it's a ListParagraph of some sort
    sectnum = mypara.Range.ListFormat.ListString

    'just in case it doesn't end in a "."
    If Right(sectnum, 1) <> "." Then
      sectnum = sectnum & "."
    End If


    'if 1 dot only then it's a parent
    dots = 0 'reset for each iteration
       'count the dots
       For j = 1 To Len(sectnum)
         If Mid$(sectnum, j, 1) = "." Then
           dots = dots + 1
         End If
       Next j

    'Find parent section
       If dots > 1 Then
         For j = Len(sectnum) - 1 To 1 Step -1
           If Mid$(sectnum, j, 1) = "." Then
             'found second "."
             parent_sec = Left(sectnum, j)
             'Look up the parent id
             parent_id = LookupParent(parent_sec)
             Exit For
```

```
          End If
        Next j
      Else
        'must be a top level
        parent_id = 0
      End If

    If mypara.Range.Bold = True Or mypara.Range.Bold = "9999999" Then '9999999 = mixed
      secttitle = CleanString(mypara.Range.Text)
    End If

    Set mynewfile = Documents.Add 'Create new document
    'mynewfile.Range.Text = CleanString(mypara.Range.Text) 'Clean String & put text in new document

    mynewfile.SaveAs FileName:=mypath & sect_id & ".RTF", FileFormat:=wdFormatRTF 'save file

    mynewfile.Close 'close file
    prevfile = sect_id & ".RTF"

    assocfile = sect_id & ".RTF" 'set filename of doc based on the section number

    'Write into database
    mystr = "INSERT INTO Word2eTariff (assoc_file, sect_id, parent_id, sect_num, sect_title) VALUES ('" & assocfile & "', " & sect_id & ", " &
parent_id & ", '" & sectnum & "', '" & secttitle & "')"
    myconn.Execute mystr

    sect_id = sect_id + 1 'Increment section ID

  Else 'append to previous doc
    If prevfile <> "" Then
      Set myfile = Documents.Open(mypath & prevfile) 'open the previous document

      myfile.Range.Text = myfile.Range.Text & mypara.Range.Text 'put text from existing document
      myfile.Save 'save the changes
      myfile.Close 'close the file
    End If
  End If
Next mypara

End Sub
Private Function LookupParent(parent_sec As String) As Integer
'This function looks up the section id of a parent section
'The return value of this function will be used for the parent_id field of a new row

Dim myrst As ADODB.Recordset 'create a recordset to store rows for look up
'execute the SELECT statement, all rows will be returned in the recorset object
Set myrst = myconn.Execute("SELECT sect_id, sect_num FROM Word2eTariff")

'Loop thru the recordset and match parent_sec with sect_num
'and return the sect_id when found.
Do While Not myrst.EOF
  If Trim(myrst!sect_num) = Trim(parent_sec) Then
    LookupParent = myrst!sect_id 'return the sect_id
    Set myrst = Nothing
    Exit Function
  End If
  myrst.MoveNext 'move to the next record
Loop

MsgBox "Error!  Could not find section id to reference the parent id."

Set myrst = Nothing 'destroy objects = a good practice
LookupParent = 0 'return 0 if an error occurs; 0 = a top level section with no parent

End Function

Private Function CleanString(mystr As String) As String

'This function strikes all the chars that are non alpha and non numeric
'from the left and the right of the string that is the current paragraph.
```

```
'It returns a clean string.

mystr = Trim(mystr) 'Trim spaces from left and right

'Clean Left side of the string
Do Until Asc(Left(mystr, 1)) > 33 And Asc(Left(mystr, 1)) < 126
  mystr = Right(mystr, Len(mystr) - 1)
Loop

'Clean Right side of the string
Do Until Asc(Right(mystr, 1)) > 33 And Asc(Right(mystr, 1)) < 126
  mystr = Left(mystr, Len(mystr) - 1)
Loop

'Return the cleaned string
CleanString = mystr

End Function
```