

## Domain Text to XML Converter

31 January 2008

Below is VBA code that will convert one coded value domain in text or MS Word format into an XML file that can be imported with the ArcFM XML Import tool in ArcCatalog.

This is very useful if you need to import a domain that can't be modeled in the physical model (e.g. a domain with unique codes but duplicate descriptions), if you want to quickly sort a domain, or if you already have a large domain in text format but don't want to key it into Visio. This can also be used to process GIS attribute data to create a domain. Additionally, large domains can be maintained and sorted alphabetically in Word and this code used to convert them into importable XML files.

The formatting rules are straightforward.

1. The name of the domain should be placed at the beginning of the domain as "Name=DomainName" <return>.
2. The type is placed next as "Type=TYPE" <return> where TYPE is either Text, Double, ShortInteger or LongInteger.
3. The domain values must be formatted DESCRIPTION<tab>CODE, with a return at the end of each line.
4. There should be no extra returns/lines between the Type=TYPE line and the DESCRIPTION<tab>CODE line.
5. The domain must not contain any duplicate codes, although duplicate descriptions are allowed.

A sample domain format is as follows:

```
Name=MyDomain
Type=Text
MyDescription1      MyCode1
MyDescription2      MyCode2
MyDescription3      MyCode3
```

Certain characters ( & ' > < " ) are perfectly legal characters inside domain descriptions in ArcCatalog. However, in an XML file these must be "escaped" using the following strings:

Find	Replace
&	&amp;
'	apos;
>	&gt;
<	&lt;
"	&quot;

To set up this tool, do the following:

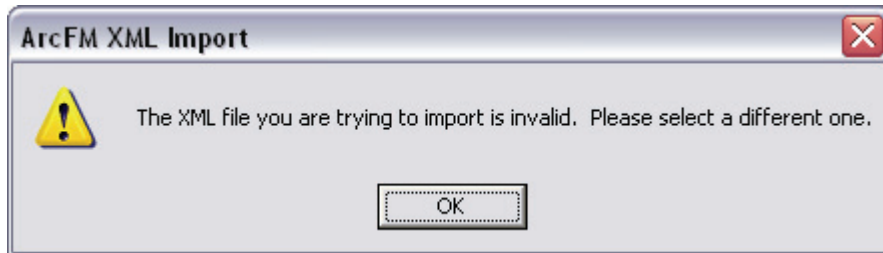
1. Start MS Word
2. Select Tools>Macro>Macros
3. Enter DomainTextToXML as the macro name and click Create
4. Select the Sub that is created (select from "Sub DomainTextToXML()" to "End Sub")
5. Paste the below code over the selected code
6. Close Microsoft Visual Basic

To use this tool, do the following

1. Start MS Word
2. Open the MS Word document that contains your domain
3. Select Tools>Macro>Macros
4. Enter DomainTextToXML as the macro name or pick this macro from the list and click Run

The domain will be saved in XML format to the current MS Word document's folder.

Note that if you have duplicate codes, unescaped characters, or extra returns/lines in your domain you will receive the following error with ArcFM 9.0 or later:



For data modeling, you will still need to create your to-be-imported domains in the physical model so that you can assign the domains in the physical model. When you go to import the XML domain, choose Overwrite to overwrite the empty domains with the contents of your XML file.

Future code enhancements:

1. Better error trapping for invalid domains.
2. Allow multiple domains in the same MS Word document.
3. Allow the user to chose the delimiter between Description and Code.
4. Allow the user to chose whether the format is Description(tab)Code or Code(tab)Description.
5. Port this to VB instead of VBA.

Here's the VBA code:

```
Sub DomainTextToXML()
'
' DomainTextToXML Macro
' John Dirkman
' 1 November 2006
'
' Future enhancements:
' 1. Better error trapping for invalid domains.
' 2. Allow multiple domains in the same MS Word document.
' 3. Allow the user to chose the delimiter between Description and Code.
' 4. Allow the user to chose whether the format is Description(tab)Code or
Code(tab)Description.
' 5. Port this to VB instead of VBA.

    Dim Msg, Style, Title, Response, DomainName, DomainType, XMLFileName,
FilePath

' Go to the start of the document
Selection.MoveUp Unit:=wdScreen

' Clear formatting
Selection.Find.ClearFormatting
```

```

Selection.Find.Replacement.ClearFormatting

' Make sure the document is properly formatted
Msg = "Welcome to the Domain Text to XML converter!" & Chr(13) & Chr(13)
& _
  "Domain Name must be formatted:" & Chr(13) & " Name=DomainName" &
Chr(13) & Chr(13) & _
  "Domain Type must be formatted:" & Chr(13) & " Type=Text or" & Chr(13)
& " Type=Double or" & _
  Chr(13) & " Type=ShortInteger or" & Chr(13) & " Type=LongInteger" &
Chr(13) & Chr(13) & _
  "Text must be formatted:" & Chr(13) & _
  " Description<tab>Code" & Chr(13) & Chr(13) & _
  "Duplicate Codes, unescaped charaters, and extra returns/lines" &
Chr(13) & _
  "will prevent the domain from being imported into ArcCatalog." &
Chr(13) & _
  "Duplicate Descriptions are allowed." & Chr(13) & Chr(13) & _
  "Only Coded Value domains can be converted." & Chr(13) & Chr(13) & _
  "Do you want to continue?" ' Define message.
Style = vbYesNo + vbQuestion ' Define buttons.
Title = "Domain Text to XML" ' Define title.
Response = MsgBox(Msg, Style, Title) ' Display message.
If Response = vbNo Then ' User chose No.
  GoTo LastLine
End If

' Get Domain Name

With Selection.Find
  .Text = "Name="
  .Replacement.Text = ""
  .Forward = True
  .Wrap = wdFindContinue
  .Format = False
  .MatchCase = False
  .MatchWholeWord = False
  .MatchWildcards = False
  .MatchSoundsLike = False
  .MatchAllWordForms = False
End With
Selection.Find.Execute

With Selection
  .MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdMove
  .MoveRight Unit:=wdSentence, Count:=1, Extend:=wdExtend
End With

DomainName = Selection.Text
'MsgBox DomainName

' Get Domain Type

With Selection.Find
  .Text = "Type="
  .Replacement.Text = ""
  .Forward = True

```

```

        .Wrap = wdFindContinue
    End With
    Selection.Find.Execute

    With Selection
        .MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdMove
        .MoveRight Unit:=wdWord, Count:=1, Extend:=wdExtend
    End With

    DomainType = Selection.Text

'   Check for correct domain type

    If DomainType <> "Text" And DomainType <> "Double" And DomainType <>
"ShortInteger" And DomainType <> "LongInteger" Then
        Msg = "Type=" & DomainType & Chr(13) & Chr(13) & "Domain Type must be
formatted:" & Chr(13) & "  Type=Text or" & Chr(13) & "  Type=Double or" &
Chr(13) & "  Type=ShortInteger or" & Chr(13) & "  Type=LongInteger" & Chr(13)
& Chr(13) & "Application exiting." ' Define message.
        Style = vbOKOnly + vbExclamation      'Define buttons.
        Title = "Domain Text to XML"         'Define title.
        Response = MsgBox(Msg, Style, Title)   'Display message.
        GoTo LastLine      'Exit
    End If

    'MsgBox DomainType
    'GoTo LastLine

'   Delete first two lines (Name and Type data)
    Selection.MoveUp Unit:=wdScreen, Extend:=wdMove
    Selection.MoveDown Unit:=wdParagraph, Count:=2, Extend:=wdExtend
    Selection.Delete

'   Required Syntax is
<CODEDVALUEELEMENT><CODENAME>Transmission</CODENAME><CODEVALUE>T</CODEVALUE><
/CODEDVALUEELEMENT>

'   1. Cull out extra hard returns.
    With Selection.Find
        .Text = "^p^p"
        .Replacement.Text = "^p"
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

'   2. Replace hard returns with tags.
    With Selection.Find
        .Text = "^p"
        .Replacement.Text = _

```

```

        "</CODEVALUE></CODEDVALUEELEMENT>^p<CODEDVALUEELEMENT><CODENAME>"
        .Forward = True
        .Wrap = wdFindContinue
    End With
    Selection.Find.Execute Replace:=wdReplaceAll

' 3. Replace tabs with tags.
With Selection.Find
    .Text = "^t"
    .Replacement.Text = _
        "</CODENAME><CODEVALUE>"
    .Forward = True
    .Wrap = wdFindContinue
End With
Selection.Find.Execute Replace:=wdReplaceAll

' 4. Clean up.
With Selection.Find
    .Text = "^p<CODEDVALUEELEMENT><CODENAME>^p"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindContinue
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text =
"<CODEDVALUEELEMENT><CODENAME></CODEVALUE></CODEDVALUEELEMENT>^p"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindContinue
End With
Selection.Find.Execute Replace:=wdReplaceAll

' 5. Add tags at the beginning of the domain.

Selection.MoveUp Unit:=wdWindow

Selection.TypeText Text:="<GXXML>"
Selection.TypeParagraph
Selection.TypeText Text:="<DOMAINS>"
Selection.TypeParagraph
Selection.TypeText
Text:="<IEPROGID>MMGxXMLImportExporterImpl.mmDomainsIE</IEPROGID>"
Selection.TypeParagraph
Selection.TypeText Text:="<DOMAIN>"
Selection.TypeParagraph

' Insert the correct domain name

Selection.TypeText Text:="<NAME>"
Selection.TypeText Text:=DomainName
Selection.TypeText Text:="</NAME>"

Selection.TypeParagraph
Selection.TypeText Text:="<DESCRIPTION></DESCRIPTION>"
Selection.TypeParagraph
Selection.TypeText Text:="<DOMAINID>999</DOMAINID>"

```

```

Selection.TypeParagraph
Selection.TypeText Text:="<TYPE>esriDTCodedValue</TYPE>"
Selection.TypeParagraph

'   Insert the correct domain type

If DomainType = "Text" Then
    Selection.TypeText Text:="<FIELDTYPE>esriFieldTypeString</FIELDTYPE>"
ElseIf DomainType = "Double" Then
    Selection.TypeText Text:="<FIELDTYPE>esriFieldTypeDouble</FIELDTYPE>"
ElseIf DomainType = "ShortInteger" Then
    Selection.TypeText
Text:="<FIELDTYPE>esriFieldTypeSmallInteger</FIELDTYPE>"
ElseIf DomainType = "LongInteger" Then
    Selection.TypeText
Text:="<FIELDTYPE>esriFieldTypeInteger</FIELDTYPE>"
End If

Selection.TypeParagraph
Selection.TypeText Text:="<MERGEPOLICY>esriMPTDefaultValue</MERGEPOLICY>"
Selection.TypeParagraph
Selection.TypeText Text:="<SPLITPOLICY>esriSPTDuplicate</SPLITPOLICY>"
Selection.TypeParagraph
Selection.TypeText Text:="<CODEDVALUEELEMENT><CODENAME>"

'   6. Add tags at the end of the domain.

charmoved = Selection.EndOf(Unit:=wdStory, Extend:=wdMove)

Selection.TypeParagraph
Selection.TypeText Text:="</DOMAIN>"
Selection.TypeParagraph
Selection.TypeText Text:="</DOMAINS>"
Selection.TypeParagraph
Selection.TypeText Text:="</GXXML>"
Selection.TypeParagraph

'   Go to the start of the document
Selection.MoveUp Unit:=wdScreen

'   7. Save the file as TXT format.

XMLFileName = DomainName & ".xml"

'MsgBox XMLFileName

ActiveDocument.SaveAs FileName:=XMLFileName, FileFormat:= _
    wdFormatText, LockComments:=False, Password:="",
AddToRecentFiles:=True, _
    WritePassword:="", ReadOnlyRecommended:=False,
EmbedTrueTypeFonts:=False, _
    SaveNativePictureFormat:=False, SaveFormsData:=False,
SaveAsAOCELetter:= _
    False

'   Get the current default folder for Word documents.
FilePath = Options.DefaultFilePath(wdCurrentFolderPath)

```

```
Msg = "Domain " & DomainName & " processed and saved as " & FilePath &
"\\" & XMLFileName
Style = vbOKOnly + vbInformation      'Define buttons.
Title = "Domain Text to XML"         'Define title.
Response = MsgBox(Msg, Style, Title)   'Display message.
```

LastLine:

End Sub