

Setup Instructions

Note: IRS standards preclude these instructions from HTML format on IRS.GOV.


- Download and install Microsoft's XML parser:
<http://msdn.microsoft.com/XML/XMLDownloads/>
 - Use either msxml6.msi or msxml6_SDK.msi (I think the "64" versions are for 64 bit operating systems). The SDK version has a nice help file that documents XML, XSLT, XPATH, and XSD
- Download and install the ReadMeF package
 - Run the setup program (double-click setup.exe)
 - Please choose a directory without spaces in the path name
 - Click the big button with the computer to start the install
 - ignore warnings about msxml4.dll or msxml6.dll if encountered
- Download and install the XSL and support files
 - Extract (unzip) the XSL to the same drive as ReadMef using the original folder names
 - Using WinZip:
 - Open the XSL2006.zip file
 - Click the toolbar "Extract" button
 - Select a drive letter
 - Select (check) "Use Folder names"
 - Click the "Extract" button (on the "popup" window)
 - Note: after the XSL is extracted, you should find a directory called "mef" containing subdirectories for the XSL and support files (images, css, and js files)
- Start ReadMef and set the default directory locations for the temporary HTML and XML "work" files, the XSL files and the XSD extract XML.
 - Access these by menu selections "Options", "Default directory locations"
 - Notes:
 - The work files (XML and XSL) are not saved unless you select menu item "Debug", "Save work files"
 - The HTML file contains document data, and should be considered sensitive "live" data. Please protect it.
 - All HTML and XML work files are deleted on program EXIT unless "Save work files" is selected.
 - The XSL work file is saved in the XSL directory location. This file is deleted after each transformation (or display using HTML) unless "Save work files" is selected.
 - The XSD extract files are optional, please see "Extracting XML from a schema for the generic display" (below)

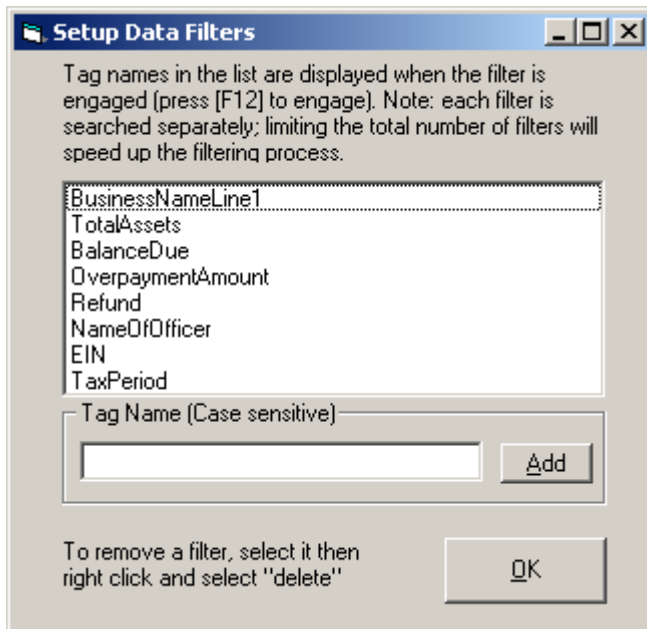
- **For push-pin support:**

- Copy xfer.html and BackArrowIE.jpg to your temporary HTML work directory (these files should be (originally) in the directory where ReadMeF was installed).
- Copy FormDisplay.js to each mef\rrdprd\sdi\versioned\yyyy\scripts directory (where "yyyy" is the tax year) (Note: this file should be in the directory where you extracted the ReadMef components) *or* add this function to each existing FormDisplay.js:

```
function SendRef(site) {  
location.href='xfer.html?RefDocID='+escape(site);  
}
```

- **Optional: Set up the Data Filters**

A data filter can provide a narrowed view of the data, displaying only those fields that you may be interested in. The filter displays only those elements with the same tag-name as those in the filter list. To use the filter click the filter button  or press F12. To set up the filter list, use menu options “Filter”, “Setup Filter” to access the following window...



Filters can be removed by right-clicking a tag name and then left-clicking “Delete”.

To add a filter (tag name), copy the tag name from either the XSD or XML (Ctrl-C) and paste it to the “Tag Name” text box, then click “Add”

Click OK to save your changes.

- **Optional: Setup an external parser**

Setting up an external XSLT processor

Using an external parser can be useful for pinpointing processor errors. The Xalan-Java XSLT processor will display the file name and line number of the instruction causing an error and makes an obvious choice as an alternative to MSXML. It is available free at <http://xml.apache.org/xalan-j/>. There are other parsers available that may be just as useful, please feel free to find what works best for you.

ReadMef Requirements for external processors

- The processor must be invoked in a DOS batch job and must be specified
- The processor takes an XML file plus an XSL file and creates an HTML file
- The DOS batch job must take three parameters; XML file, XSL file, and HTML file. These parameters can be in any order but the order must be specified.
- Transform options are specified via menu “Options”, “XML parser”, “Setup External Parser”

Recipe for using the Xalan-Java XSLT processor (this is just one way to set it up)

- Download the xerces-j distribution package (Xerces-J-bin.2.7.1.zip – or version updated name)
- Unzip or Extract the files to a directory of your choice (check user folder names)
Note: the example below uses “D:\mef_old\Xerces”)
- If you don’t have it already, download and install the Java runtime environment (JRE) or Java development kit (JDK) (available at <http://java.sun.com>)
- Build three batch jobs
 - SetPath
 - indicates where “java.exe” lives
 - SetClassPath
 - sets the classpath variable
 - Transform
 - invokes SetPath, SetClassPath and calls the java class

- Example batch jobs (using “D:\mef_OLD\Xerces” as location for xalan and the batch files)
 - setPath5a.bat (SetPath) points to the location of “java.exe”
 - `set PATH=%path%;D:\Jdev\jdk\bin`
 - setclass.bat (SetClassPath) lines up the Xalan class objects (all on one line)
 - `set classpath=D:\mef_OLD\Xerces\xalan-j_2_7_0\xalan.jar;D:\mef_OLD\Xerces\xalan-j_2_7_0\serializer.jar;D:\mef_OLD\Xerces\xalan-j_2_7_0\xml-apis.jar;D:\mef_OLD\Xerces\xalan-j_2_7_0\xercesImpl.jar`
 - xform.bat (Transform) with parameters in 1) XML, 2) HTML, 3) XSL order
 - `cd D:\mef_OLD\Xerces`
 - `IF "%CLASSPATH%"==" " (`
`call D:\mef_OLD\Xerces\SETCLASS`
`call D:\mef_OLD\Xerces\SETPATH5a`
`)`
 - `java org.apache.xalan.xslt.Process -IN %1 -XSL %3 -OUT %2`
- Specify the parser options
 - Start ReadMef
 - Choose menu options “Options”, “XML Parser”, “Setup External Parser”
 - Locate your batch job using the drive, path, and file windows
 - Arrange the order of parameters to match your batch job
 - For my example, %1 = XML, %2 = HTML, and %3 = XSL
 - Click “Accept”
- To use the external parser choose menu options “Options”, “XML Parser”, “External Batch Parser”

- **Optional:** Setup "KeyDataFields.txt" to see extracted data on the "tree-view" nodes (table of contents) in ReadMef

Key Data Fields

Key pieces of data can be displayed as part of document names in the tree-view pane to make it easier to identify repeating documents. The program uses an ASCII text file in the program's directory named "KeyDataFields.txt" to identify this association. A simple version of this file is included during the program setup. KeyDataFields.txt is an old-fashioned (positionally based) data file with the following format:

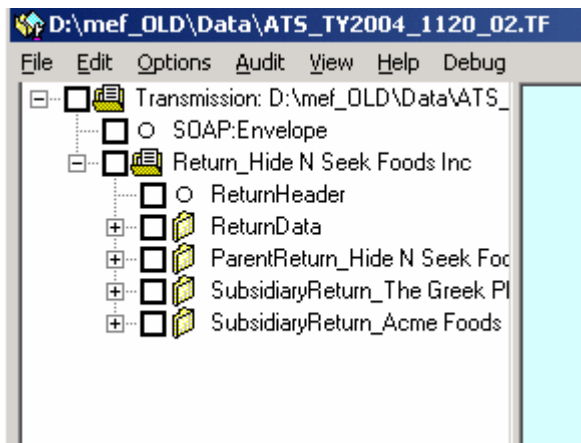
- Document tag value (columns 1-34)
- relative XPath reference (columns 35 to end of line)
 - XPath is relative to the document element in columns 1-34
 - XPath may contain attribute references (use @ sign before attribute name)
- No comments allowed
- Works for any "document" under element "ReturnData", as well as "Return", "ParentReturn", and "SubsidiaryReturn" elements.

The data is extracted and stored in the same directory as the index file (data file name + ".ix"), and should be guarded with the same level of security as your data files.

Example KeyDataFields.txt contents (XPath reference starts in column 35):

Return	ReturnHeader/Filer/Name/BusinessNameLine1
ParentReturn	ReturnHeader/ParentCorp/Name/BusinessNameLine1
SubsidiaryReturn	ReturnHeader/SubsidiaryCorp/Name/BusinessNameLine1
IRS926	TransferorIdentificationNumber
IRS1118	CorporationEIN
IRS1118AMT	CorporationEIN
IRS1118ScheduleI	ForeignOilGasExtIncome/ForeignCountry
IRS4562	BusinessOrActivity
IRS5471	@documentId
IRS5471ScheduleJ	NameOfBusinessFilingForm5471/BusinessNameLine1

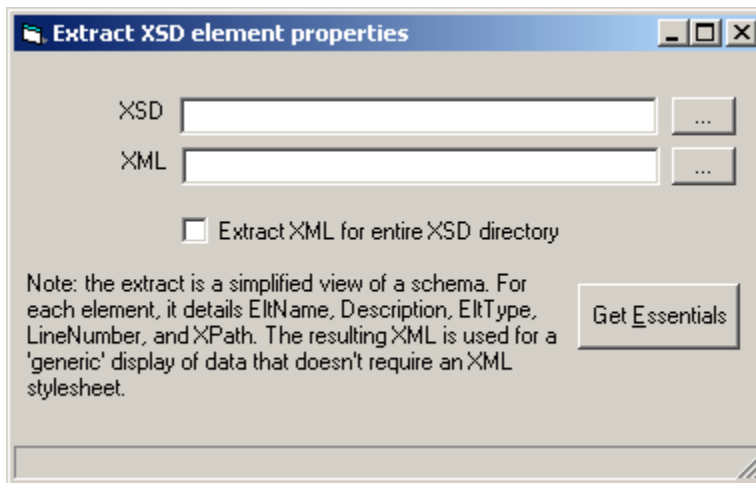
Example display using key data fields:



Extracting XML from a schema for the generic display

The generic display uses an extract of an XSD schema (if available) to display the line number and description for data elements. The extract is a simple XML file listing the valid elements within a return document. Using the extract speeds up the look-up process that would otherwise involve resolving all referenced element groups and sub-group definitions within an XSD. For each return document element Element-name, Description, Element-Type, Line Number and XPath are captured. If "LineNumber" and "Description" elements are not included in the XSD (an IRS standard, but not an XML standard), the values for these are empty. To see what the extract looks see "XSD Extract" below.

You can update the extracted XML via an included utility available with menu selections "Options", "Default directory locations", "Build XSD extract XML". You should see a window that looks like this:



To build the XML extract for one return document XSD, locate the XSD (by clicking the "..." button), name the XML extract file with the same name, but with an extension of "xml" instead of "xsd". (The XML file should be in a different directory than the XSD).

To build XML extracts for all the XSD in one directory, make sure "Extract XML for entire XSD directory" is checked prior to locating the XSD.

Click "Get Essentials" to generate the XML

Note: ReadMef looks for the extract XML in the directory specified by menu items "Options", "Default directory locations", "XSD extract location"

XSD Extract (example)

"Extract" XML for a few fields of form 3115

```
<?xml version="1.0" ?>
- <EssentialElts>

- <ElementDescription>

  <EltName>OtherAccountingMethod</EltName>
  <Description>Other accounting method</Description>
  <EltType>CheckBoxType</EltType>
  <LineNumber>Part I Line 1b</LineNumber>
  <XPath>IRS3115/OtherAccountingMethod</XPath>
</ElementDescription>
- <ElementDescription>
  <EltName>IsAcctMthdChangeNotApply</EltName>
  <Description>Is accounting method change one for which scope
    limitations of Sec 4.02 Rev Proc 2002-9 do not
    apply?</Description>
  <EltType>BooleanType</EltType>
  <LineNumber>Part I Line 2</LineNumber>
  <XPath>IRS3115/IsAcctMthdChangeNotApply</XPath>
</ElementDescription>
- <ElementDescription>
  <EltName>IsTYOfChangeRqrTakeEntireAmt</EltName>
  <Description>Is tax year of change the final tax year of trade or
    business for which TP would be required to take entire
    amount of Sec 481(a) adjustment into account in computing
    taxable income?</Description>
  <EltType>BooleanType</EltType>
  <LineNumber>Part I Line 3</LineNumber>
  <XPath>IRS3115/IsTYOfChangeRqrTakeEntireAmt</XPath>
</ElementDescription>
- <ElementDescription>
```

...