Goal: The purpose of this tool is to allow easier transition between transforming XML resources into EAD in order to facilitate the import into ArchivesSpace

Once you get used to the process and setup this process will save you time as most of the information is imported for you. It is only a little long and detailed to ensure correct imports.

You will need:

1. Access to some XSL transformation tool (This tutorial has been written for Oxygen XML Editor)
2. Excel
3. xml2EAD.xsl file packaged with this spreadsheet

Steps:

1. **Setup Oxygen** 
   1. If done correctly, you will only need to **do this once** for your transformation needs as long as you are using the same Oxygen setup
   2. Open Oxygen, set up a transformation:
      1. You can **follow** this **tutorial**: <https://icantiemyownshoes.wordpress.com/2014/03/04/tutorial-how-to-turn-a-spreadsheet-into-the-contents-list-of-an-ead-encoded-finding-aid/>
      2. **Or** open up the transformation tool (the wrench icon in the tool bar), Document > Transformation > Configure Transformation Scenario, or **CTRL+SHIFT+C**
      3. Create a **new** XML transformation **with XSLT**.
      4. **Set XML URL** to ${currentFileURL} (\*if you have saved your xml import discussed later it will default to your file name, if not you can choose what your file name should be)
      5. **Set XSL URL** to where ever you saved the xml2EAD.xsl file (you can use the browse function)
      6. **Select** a transformer (Saxon-PE is a safe choice)
      7. Check **OK**
   3. New Excel with Oxygen (**ERROR** MAY OCCUR IF YOU DON'T DO THIS)
      1. Chances are you have a newer version of excel (2007 or Newer) that is incompatible with Oxygen. In which case, follow this neat Oxygen tutorial <https://www.oxygenxml.com/doc/versions/18.1/ug-editor/topics/import-excel-new-format.html>
2. **Excel Sheet**
   1. This excel file contains a **Template tab** to use for the transformation (don't use your own unless you know what you're doing, the excel **headers are formatted** to work with the xml2EAD.xsl file). Just copy that to a new Excel sheet, and start your resource (but **keep reading there is more useful information**).
   2. This excel file also contains an Example template (use this, seriously), which we will discuss things you need to consider when creating a resource (**THIS PART IS IMPORTANT FOR A GOOD TRANSITION INTO ARCHIVESSPACE**)
      1. You must include at **ONE** collection in at least one row. (Keep the collection\_level **lowercase**)
      2. The **association\_id** will let you **associate** series with subseries and files (This transformer can only go down two levels, series > subseries > files )
         1. **Keep Series and Subseries <= 9**
         2. Association\_id indicates collection level
         3. Begin the subseries with the starting number of the series.
            1. (i.e if Series was 2 then subseries can be 20, 21,..29)
         4. Begin the files you want under the subseries with the same as the subseries
            1. (i.e if subseries was 20 then files are all 20 20 20 20
            2. (if it was 25 then they would be 25 25 25)
      3. **Use** **machine readable values if you don't want thousands of controlled** **values** in ArchivesSpace for: extent\_type, data\_type, instance\_type, container\_type\_(1,2,3).
         1. if you're wondering what this means: make sure your values are all lowercases and spaces are changed to underscores. (Use the example as a template)
      4. **Add** date\_begin and/or date\_end. It will format your transformation properly. If not it will use the date\_expression and you will find yourself having to fix it anyways.
3. **Transformation**
   1. UNLV uses some default values, but they can be changed at the top of xml2EAD.xsl (changed their values as needed)
   2. Once you have typed up your resource in XML, import it to Oxygen
      1. File > import > MS Excel FILE
      2. Browse for your Excel file
      3. Choose a tab if you have multiple sheets
      4. **Select First row contains field names** (it will NOT work without this option)
      5. Click import
      6. Optional (\*Save this new file created if you want to use your transformation variable you added in the first step)
   3. Transform
      1. Click on the red play button next to the wrench where you did your transformation (**while having the tab of your import currently selected**). Same thing under Document > Transformation > Apply Transformation Scenario(s), or **CTRL+SHIFT+T**
      2. Select the desired transformation you created
      3. An EAD record will be created. **REVIEW BEFORE IMPORTING.** 
         1. Things to look for:
            1. Make sure values mentioned above are machine readable
            2. Make sure your dates are correct
            3. Look for red underlines (usually indicates an error in your EAD) use Oxygen tool for explanation.

Disclaimer, some maybe local rules and do not follow EAD Formats. I.E oversized\_box

* + 1. Make sure you save it where you can find it to import to ArchivesSPace

1. **Importing to ArchivesSpace**
   1. Login as a user that can import (check with your administrator)
   2. Click Create > Background Jobs
   3. Select Import Data for Job Type
   4. Select EAD for Import Type
   5. Add file
      1. Select your EAD file that you created
      2. Multiple files can be imported at one time
   6. Things to check after import
      1. Depending on your library you may need to spread out the unique identifier across the id fields. It is ArchivesSpace design decision to import this way (https://github.com/archivesspace/archivesspace/pull/338)
      2. Did everything get imported correctly?
      3. Are your notes correct?
      4. Is the order of your series correct?
      5. Are your documents published?
      6. Did you accidently create non machine readable values? (check your excel sheet again for this) If so record these, so they can be merged with their correct values.
      7. Overall, its good?
      8. If you answered yes to these questions then you have successfully imported your XML to EAD to ArchivesSpace