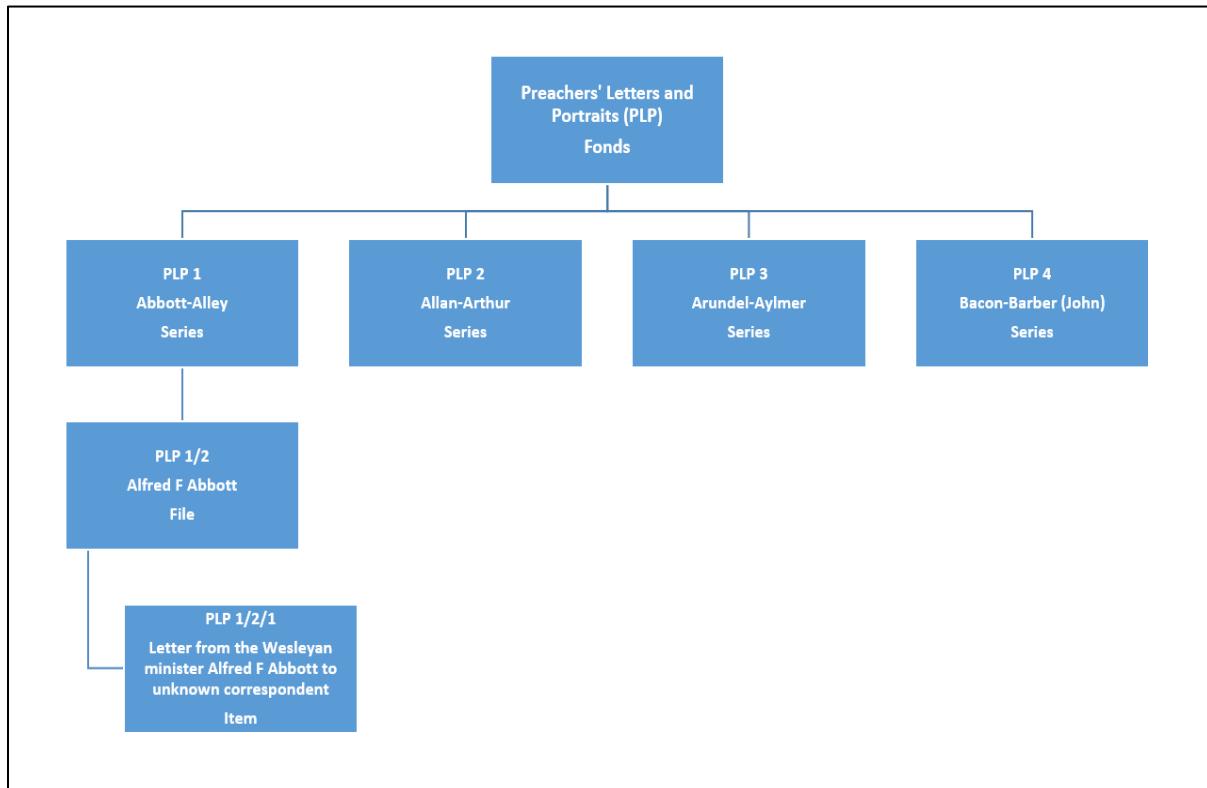


## Excel to EAD converter

### Part 2

#### **Script 1**

1. Script 1 is designed for archives structured as Series → File → Item, e.g.



Each component is expected to follow a hierarchical <united> pattern:

- Series: PLP 1
- File: PLP 1/1
- Item: PLP 1/1/1

2. The script reads a spreadsheet (.xlsx) with columns for level, unitid, unittitle, unitdate, extent, and scopecontent and builds a nested EAD XML tree. The column headings are expressed in the following way:

- <c level>
- <unitid>
- <unittitle>
- <unitdate>

- <extent>
- <scopecontent>

The spreadsheet template looks like this:

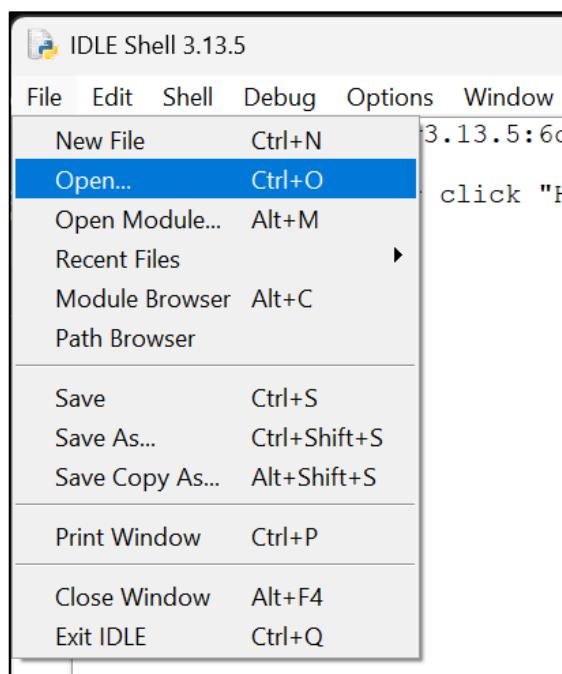
	A	B	C	D	E	F
1	<c level>	<unitid>	<extent> </extent>	<unittitle></unittitle>	<scopecontent>	<unitdate>
2	Series	PLP 1	661 items	Abbott - Alley		1745-1970
3	File	PLP 1/1	4 items	Fragments/Unidentified material		22 April 1845; [1846]; 27 October 1863
4	File	PLP 1/2	3 items	Alfred F Abbott (1816-1879)		
				Addressed to 'Rev[eren]d and dear Sir'. Written at Whitehaven, Cumberland.		
5	Item	PLP 1/2/1	1 item	Letter from the Wesleyan minister Alfred F Abbott asks whether he should mention at the District Meeting or at the Conference that he has 'received no part of a young married man's allowance, when two young married men in this District, have received the allowance'. He writes that 'by some means or other I have been left out of the married list'. Addressed to 'My Dear Bro[the]r'. Written at Wansbeck Terrace, Morpeth.		22 April 1845

3. The script omits blank fields and supports multiple paragraphs in <scopecontent>.

4. To use the script, place the spreadsheet(s) and the script together in a new folder:

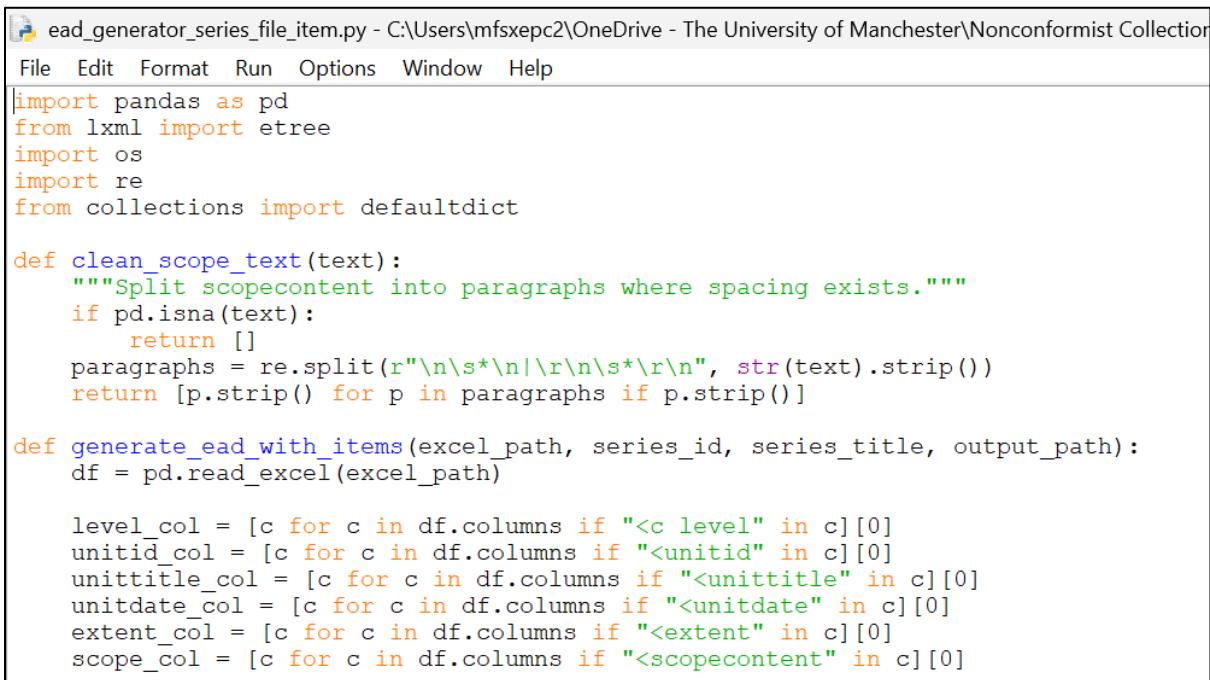
Name	Status	Date modified	Type	Size
ead_generator_series_file_item	✓	21/10/2025 16:37	Python File	4 KB
PLP 1	✓	21/10/2025 16:32	Microsoft Excel Work...	45 KB
PLP 2	✓	21/10/2025 16:36	Microsoft Excel Work...	37 KB

5. Open IDLE, click on 'File' and select 'Open':



6. Open the Python script saved in the new folder:

Name	Status	Date modified	Type	Size
ead_generator_series_file_item.py	✓	21/10/2025 16:37	Python File	4 KB



```

ead_generator_series_file_item.py - C:\Users\mfsxepc2\OneDrive - The University of Manchester\Nonconformist Collection
File Edit Format Run Options Window Help
import pandas as pd
from lxml import etree
import os
import re
from collections import defaultdict

def clean_scope_text(text):
    """Split scopecontent into paragraphs where spacing exists."""
    if pd.isna(text):
        return []
    paragraphs = re.split(r"\n\s*\n|\r\n\s*\r\n", str(text).strip())
    return [p.strip() for p in paragraphs if p.strip()]

def generate_ead_with_items(excel_path, series_id, series_title, output_path):
    df = pd.read_excel(excel_path)

    level_col = [c for c in df.columns if "<c level" in c][0]
    unitid_col = [c for c in df.columns if "<unitid" in c][0]
    unittitle_col = [c for c in df.columns if "<unittitle" in c][0]
    unitdate_col = [c for c in df.columns if "<unitdate" in c][0]
    extent_col = [c for c in df.columns if "<extent" in c][0]
    scope_col = [c for c in df.columns if "<scopecontent" in c][0]

```

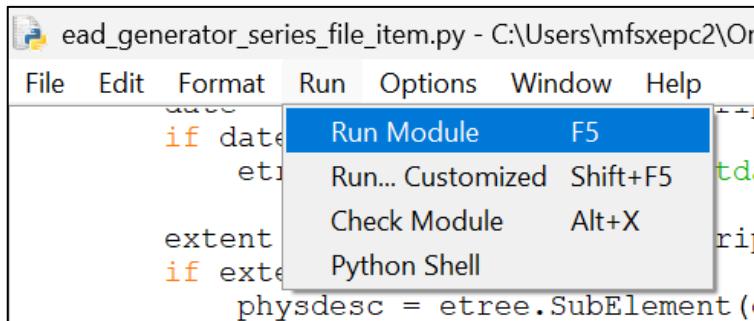
7. Amend the last line in the script so that it can identify the spreadsheet sitting in the same folder:

```
#  Final function call – edit this for each series
generate_ead_with_items("PLP 1.xlsx", "PLP 1", "Series: PLP 1", "output/PLP_1_output.xml")
```

In this example, we are converting PLP 1:

- Excel file name ("PLP 1.xlsx")
- <unitid> (usually a short ID like "PLP 1")
- <unittitle> (title of your series or collection)
- Output filename

8. Run the script by pressing F5 or selecting ‘Run Module’:



9. If successful, you will see receive the following message:

```

IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> = RESTART: C:\Users\mfsxepc2\OneDrive - The University of Manchester\Nonconformist Collections\Cataloguing\Cataloguing Projects\Cataloguing Process\EAD\Experiments\Conversion tool\ead_script_1_S-F-I\ead_generator_series_file_item.py
     XML successfully saved to: output/PLP_1_output.xml
>>>

```

10. Return to the folder you created at step 4. Here, you should see the spreadsheet(s) and script saved earlier, but now they are joined by a folder titled ‘output’, which is created automatically by the script:

Name	Status	Date modified	Type	Size
output	✓	21/10/2025 17:32	File folder	
ead_generator_series_file_item	✓	21/10/2025 17:25	Python File	4 KB
PLP 1	✓	21/10/2025 16:32	Microsoft Excel Work...	45 KB
PLP 2	✓	21/10/2025 16:36	Microsoft Excel Work...	37 KB

11. Inside the folder is the newly created xml file:

Name	Status	Date modified	Type	Size
PLP_1_output	✓	21/10/2025 17:32	Microsoft Edge HTM...	225 KB

```

▼<ead>
  ▼<archdesc level="collection">
    ▼<dsc>
      ▼<c level="series">
        ▼<did>
          <unitid>PLP 1</unitid>
          <unittitle>Series: PLP 1</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian"/>
        </did>
      ▼<c level="file">
        ▼<did>
          <unitid>PLP 1/1</unitid>
          <unittitle>Fragments/Unidentified material</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian">n/a</unitdate>
        ▼<physdesc>
          <extent>4 items</extent>
        </physdesc>
        </did>
      </c>
      ▼<c level="file">
        ▼<did>
          <unitid>PLP 1/2</unitid>
          <unittitle>Alfred F Abbott (1816-1879)</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian">22 April 1845; [1846]; 27 October 1863</unitdate>
        ▼<physdesc>
          <extent>3 items</extent>
        </physdesc>
        </did>
      ▼<c level="item">
        ▼<did>
          <unitid>PLP 1/2/1</unitid>
          <unittitle>Letter from the Wesleyan minister Alfred F Abbott to unknown correspondent</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian">22 April 1845</unitdate>
        </did>
      </c>
    </dsc>
  </archdesc>
</ead>

```

12. To run the script on another file, amend the final line:

```
#  Final function call – edit this for each series
generate_ead_with_items("PLP 2.xlsx", "PLP 2", "Series: PLP 2", "output/PLP_2_output.xml")
```

```
= RESTART: C:\Users\mfsxepc2\OneDrive - The University of Manchester\Nonconformist Collections\Cataloguing\Cataloguing Projects\Cataloguing Process\EAD\Experiments\Conversion tool\ead_script_1_S-F-I\ead_generator_series_file_item.py
 XML successfully saved to: output/PLP_2_output.xml
>>>
```

Name	^	Status	Date modified	Type	Size
 PLP_1_output		<input checked="" type="checkbox"/>	21/10/2025 17:32	Microsoft Edge HTM...	225 KB
 PLP_2_output		<input checked="" type="checkbox"/>	21/10/2025 17:48	Microsoft Edge HTM...	139 KB

```
▼<ead>
  ▼<archdesc level="collection">
    ▼<dsc>
      ▼<c level="series">
        ▼<did>
          <unitid>PLP 2</unitid>
          <unittitle>Series: PLP 2</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian"/>
        ▲</did>
      ▼<c level="file">
        ▼<did>
          <unitid>PLP 2/1</unitid>
          <unittitle>John Allin</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian">13 April 1847</unitdate>
        ▼<physdesc>
          <extent>2 items</extent>
        ▲</physdesc>
        ▲</did>
      ▼<c level="item">
        ▼<did>
          <unitid>PLP 2/1/1</unitid>
          <unittitle>Letter</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian">13 April 1847</unitdate>
        ▼<physdesc>
          <extent>1 item</extent>
        ▲</physdesc>
        ▲</did>
      ▲</c>
      ▼<c level="item">
        ▼<did>
          <unitid>PLP 2/1/2</unitid>
          <unittitle>Letter</unittitle>
          <unitdate datechar="creation" era="ce" calendar="gregorian">No date</unitdate>
        ▼<physdesc>
```