Anatomy of an ORA Autotype template

During a recent ORA group zoom session, one question led to over an hour and a half discussion about writing autotype templates, and the different methods used by various members of the group.

As most of the group members are new to ORA, ie non-programmer beginners with less than 6 months experience, some with more than 6 months, and a few fairly experienced members, some of the discussion 'went right over the heads' of most of the beginners. This document is my attempt to explain a common generic template I am constructing, and how and why it works. I am going to make a small assumption, rightly or wrongly, that the reader has had a look at ORA, has the basic understanding of what it can do, and maybe has ventured into the world of scripting Text and Auto Type templates.

If you are unfamiliar with the layout of the ORA Settings page, it may be better to review it first, from page 21, and then come back to the beginning.

Table of Contents

ORA Settings

From an ORA user's point of view, there are a few things we should remember about ORA.

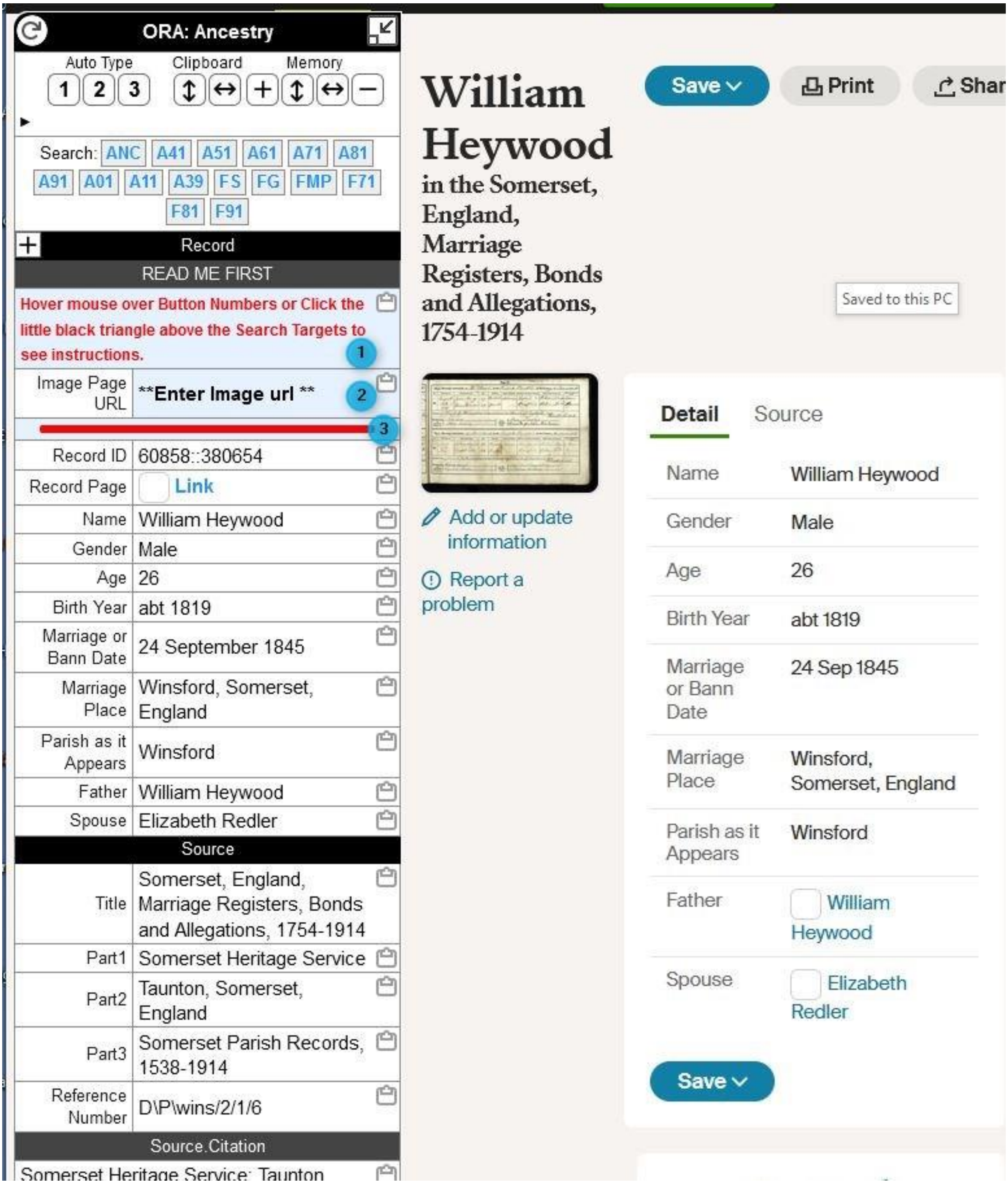
1. ORA does not know anything about your target application, how it works, what data it already contains in fields or even how to navigate around that application. We have to tell it what to do, and if we get it all correct, it will do it extremely fast and the same way every time we click the Autotype 'run' button. ORA cannot tell if it was successful, we have to check that.
2. We should, however, tell ORA what our application is going to be, and if it is one of the ***Supported Applications*** as at ORA version 1.70
3. , it will check that that application is running. If you specify say {target=fh} on the first line of an autotype template, it will warn you if it cannot find fh.exe in the running applications list when you click the ***Show Current Targets*** button.
4. More than 99% of the time, ORA templates will run exactly the same on any of the four supported web browsers: Firefox, Chrome, Opera and Edge. You can export all your current ORA development by clicking the Export Settings on one browser, and import the settings file into another browser, even if it is on another system that has a valid ORA license. We can also export and import a current Collection in a similar way.
5. In a simplified explanation, ORA stores the information from the websites you visit, which it scrapes for html data tables and content, in a json database, by Service, Collections within the Service, Records within a Collection and Templates within each Record. A Service can have multiple Collections, and each Collection can have multiple Records, and each Record can have multiple Templates. For example, Ancestry is a service; England Births and Baptisms, England Deaths and Burials, 1841 UK Census, 1851 UK Census, etc, are Collections of records; each 1851 Census enumeration page is a separate record, and we may write more than one text template and auto type template for each Census page we visit.
6. ORA fields are enclosed by square brackets, ie [fieldname].
7. ORA navigation commands or keyboard keywords are enclosed by curly brackets, ie {Ctrl+space}, {up}, {enter}. A list of all the current Commands is available at auto-type-keyboard-keywords . They can be entered in upper or lower case. These are interpreted by the target application as if you had pressed those keystrokes on your computer keyboard.
8. Conditional expressions and conditional fields are enclosed by chevrons, ie <[fieldname]>, and can be nested, ie <[fieldname], <[another field]>> etc. in these circumstances, if [fieldname] does not have any value, it will not be output.

9. You can assign a value to an existing fieldname or a new fieldname using the [=:fieldname:value] or [==:fieldname:value] The double ==: variant makes it a persistent variable, ie present in memory until you move away in the browser to a new record. Ie [==:EventType:Marriage] To force a clear of a variable you can leave the value part blank, ie [=:EventType:]
10. You can test that the [fieldname] has a value, and if true, perform an action. For example, <[?:Age]He was [Age] years old at the time>. If Age has a value, say 20. then the output would be "He was 20 years old at the time". If [Age] has no value, ie blank, nothing would be output.
11. You can test that the [fieldname] has a specific value, and if true, perform an action. For example <[?:Marital Status=Married]Marital Status: Married>, or <[?:Gender=Male]Child was a boy>
12. You can do various transformations on Fields, such as [Name:nameSurname], [Name:nameGiven:replace:.::l] This will take a Name field [Name] which contains say "Winston S. Churchill", output the Surname first, then a comma and space, and then Winston S The dot after the S was removed with the :replace:.::l (:replace:*replace what*:*with what*:and a lower case L to signify Literal mode) There are many types of transforms that you can do – see the Transforms section on the ORA help contents.
13. You can pass a FieldName or multiple FieldNames as parameters to a Library Template. Ie [lib.libraryname:FieldName1:FieldName2] In the library template, these parameters can be interpreted as {1} which will use the literal value of the first parameter FieldName1, or as [{1}] to mean the content value of the FieldName1; {2} as the literal value of the second parameter FieldName2, [{2}] as the content of the second parameter FieldName2. etc

You will see the application of some these concepts in the next sections in the Auto Types and Library templates

Anatomy of my Generic BMD AutoType template .

William Heywood married an Elizabeth Redler in Somerset in about 1845. A search on *Ancestry* (the Service) finds the record in the Collection called *Somerset, England, Marriage Registers, Bonds and Allegations, 1754-1914* at https://www.ancestry.co.uk/discoveryui-content/view/380654:60858



Hover mouse over Button Numbers or Click the little black triangle above the Search Targets to see instructions.

Image Page URL	**Enter Image url **
Record ID	60858::380654
Record Page	Link
Name	William Heywood
Gender	Male
Age	26
Birth Year	abt 1819
Marriage or Bann Date	24 September 1845
Marriage Place	Winsford, Somerset, England
Parish as it Appears	Winsford
Father	William Heywood
Spouse	Elizabeth Redler
Source	
Title	Somerset, England, Marriage Registers, Bonds and Allegations, 1754-1914
Part1	Somerset Heritage Service
Part2	Taunton, Somerset, England
Part3	Somerset Parish Records, 1538-1914
Reference Number	D\P\wins/2/1/6
Source.Citation	
Somerset Heritage Service; Taunton	

William Heywood
in the Somerset, England, Marriage Registers, Bonds and Allegations, 1754-1914

Name	William Heywood
Gender	Male
Age	26
Birth Year	abt 1819
Marriage or Bann Date	24 Sep 1845
Marriage Place	Winsford, Somerset, England
Parish as it Appears	Winsford
Father	William Heywood
Spouse	Elizabeth Redler

This analysis will be on my use of my Generic BMD Autotype template in this collection, with the Text Templates, Auto Type templates and Library Templates used. I do Export this AutoType to a file, and import it into a new Collection when necessary. Note that there are only 3 Text Templates showing (there are 7) and that I have 5 Auto Type buttons; Process the Marriage data; optionally add his Occupation and Residence facts, and test for the presence of a Source in Family Historian.

The Text Templates

1. READ ME FIRST ①

   ```
   <\<span style="color:Red; font-size: 80%;font-weight: bold"\>
   Hover mouse over Button Numbers or Click the little black triangle above the
   Search Targets to see instructions.
   \</span\>>
   ```

   This is my reminder to check instructions, In this part of the Text Templates you can use html but as the chevrons are 'reserved' characters in ORA, they need to be 'escaped' with a \ so the ORA interpreter passes them through ok.

2. Image Page URL ②

   ```
   # Ancestry
   <[?:o.Service=Ancestry]
   <[Image Page URL:split:?use:1]|**Enter Image url **>
   >
   #Findmypast
   <[?:o.Service=Findmypast]
   <[Image Page URL:split:&parent:1]|**Enter Image url **>
   >
   #FamilySearch
   <[?:o.Service=FamilySearch]
   <[Image Page URL:split:&pers:1]|**Enter Image url **>
   >
   ```

   With these three Services, I have examined the URLs of the services to determine which parts will actually take me to the Image record correctly, and what trailing parts can be discarded, as they are normally the web browsers way of getting back to the previous screen when you click the back arrow on the browser. I have utilised the split transform [fieldname:split:where to create the first split:which part do I want].   We can test the Service field for content <[?:o.Service or <?:Page.Service and if it equals the Service that we are using, process the following fields, etc, up to the closing > Note that on Ancestry I have split on the ?use; on Findmypast I have split on the &parent and FamilySearch on &pers, and I want to retain the first part before the split as indicated by the :1

   If you use other Services, like MyHeritage, Trove, etc, you can easily add another little section into this Text Template.

3. var.Set Event Flags
   <[var.Set Event Flags]|** Set Event Flags y/n **>
   This and the next 'prompt' Text Templates do not show on the ORA Control Panel, because they have the prefix var. We can only prompt for them within an Auto Type Template that tests for them. This one takes a Y or N to help the decision making in the AutoType whether to process a Library Template, or skip to the next. *See AutoType 1*

4. var.Update Fact Data
   <[var.Update Fact Data]|** Update Fact Date and Place y/n **>
   Again this Text Template is not shown on the ORA Control Panel. I use the Y or N response to determine whether or not to overwrite any existing data that exists already in Family Historian. You may have a proven actual date and place already, and do not want this AutoType to overwrite it. If this is the case, then by not overwriting, we can still Source the entry, if we did not have one before. *See AutoType 1*

5. var.Their Occupation
   <[var.Their Occupation]|** Enter their Occupation from Image **>
   This information has not been transcribed and therefore is not available on the active web page, but is available from directly reading the information off the image. We can therefore prompt for it during the execution of an Auto Type Template, and this is a convenient way of adding the data. *see AutoType 2*

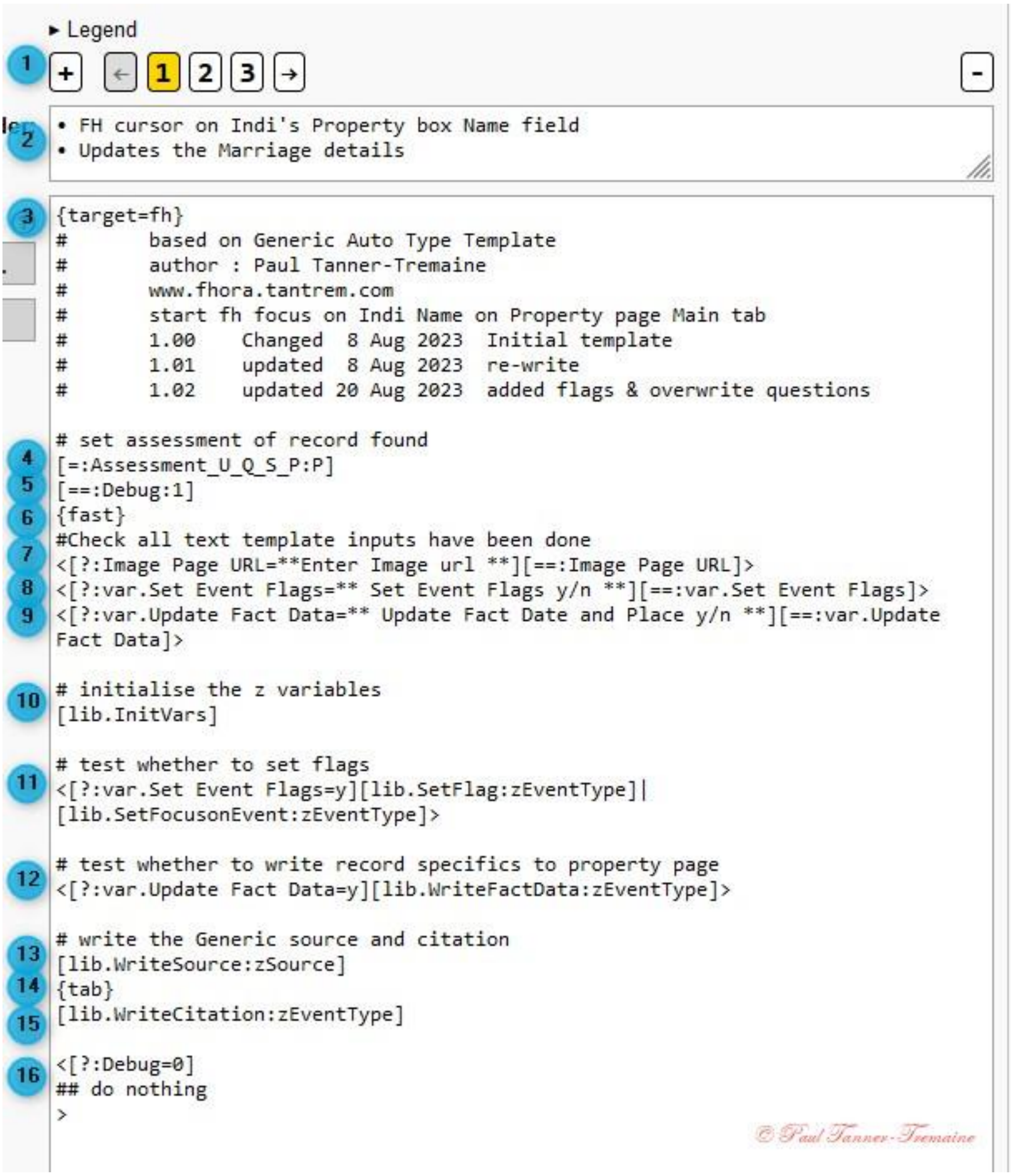
6. var.Their Residence
   <[var.Their Residence]|** Enter their Residence from Image **>
   Same comment as for Text Template 5. *see AutoType 3*

7. =
   [lib.DividerBar]
   This is to show that we can use Library Templates within Text Templates as well as Auto Type Templates. This is the text template that shows the red horizontal bar in the ORA Control Panel. By having it as the last of the text templates, it identifies which user fields need to be completed above the 'scraped' fields that ORA has found. Once we have entered data into the prompt, that Text Template becomes 'active' and will be shown below the bar. The Library Template consists of:

```
# creates a divider bar in the Control Panel
# Call as: [lib.DividerBar]
# Use in the first or last Text Template, and enter only an equal sign in the Header field of the Template
\<hr style='border: 3px solid Red; border-radius: 2px; margin:4px 15px 4px 15px'\>
```

The Auto Type Templates

Template button 1



```
• FH cursor on Indi's Property box Name field
• Updates the Marriage details
```

```
{target=fh}
#       based on Generic Auto Type Template
#       author : Paul Tanner-Tremaine
#       www.fhora.tantrem.com
#       start fh focus on Indi Name on Property page Main tab
#       1.00    Changed  8 Aug 2023  Initial template
#       1.01    updated  8 Aug 2023  re-write
#       1.02    updated 20 Aug 2023  added flags & overwrite questions

# set assessment of record found
[=:Assessment_U_Q_S_P:P]
[==:Debug:1]
{fast}
#Check all text template inputs have been done
<[?:Image Page URL=**Enter Image url **][==:Image Page URL]>
<[?:var.Set Event Flags=** Set Event Flags y/n **][==:var.Set Event Flags]>
<[?:var.Update Fact Data=** Update Fact Date and Place y/n **][==:var.Update
Fact Data]>

# initialise the z variables
[lib.InitVars]

# test whether to set flags
<[?:var.Set Event Flags=y][lib.SetFlag:zEventType]|
[lib.SetFocusonEvent:zEventType]>

# test whether to write record specifics to property page
<[?:var.Update Fact Data=y][lib.WriteFactData:zEventType]>

# write the Generic source and citation
[lib.WriteSource:zSource]
{tab}
[lib.WriteCitation:zEventType]

<[?:Debug=0]
## do nothing
>
```

This is the complete Auto Type Template for Button 1. It processes the data from the Ancestry website page and captured in the ORA Control Panel, as shown at the start of this *anatomy*, and updates Family Historian with the relevant information. Most of the functionality is in common library templates, which makes it ideal for being a 'generic' template that can be copied with minimal changes. I'll try to explain each line by the blue numbering, and the reader can skip to the *library templates* section where necessary.

1. This is the familiar row where one can select or add a new template, with the clickable Legend above which will show all the Reminder lines. I have kept this Auto Type to 5 options : 1 to add all the Marriage details; 2 to optionally add a Residence fact; 3 to add an optional Occupation fact; check for Source and 5 Add the Source.

2. This is the Reminder for button 1.

3. The **{target=fh}** specifies to ORA that the target for the output is Family Historian.
   The next seven lines are all commented out by the # in the first position giving general information about the template.

4. **[=:Assessment_U_Q_S_P:P]** This is creating a new variable Assessment_U_Q_S_P with a value of P for Primary. The Assessment: is used to evaluate the Source/Citation that is being made, and this field is used by the library template, [lib.SetAssessment] which is called in the [lib.WriteSource] library template. The options that I could use are Unreliable, Questionable, Secondary or Primary. The [lib.SetAssesment] tests for the value and applies the appropriate number of [down]'s to apply in the Assessment drop down.

5. **[==:Debug:1]** Here I am assigning a field Debug with the value of 1. Why? Well, if you look down the autotype to the rows at point 16, I have 3 rows which test if Debug=0, which will fail and therefore any rows of code within that test will not be executed. If I had problems in writing the autotype template, I can move, not copy, the **<[?:Debug=0]** up to say the row below the [lib.InitVars], and then click the Test and possibly the trace, and the test will stop after the InitVars. If that is all ok, I can then move(not copy) the **<[?:Debug=0]** to below the flag setting section and see if that runs ok. If not, then I know that the error must be in that section. I keep on moving the **<[?:Debug=0]** over the sections, until I have the **<[?:Debug=0]** back at the bottom of the auto type template and all runs fine!

6. **{fast}** or **{slow}** to begin with. This tells the ORA interpreter to run either in a slow mode or in a fast mode. I would only use {fast} when I am confident that my Auto Type template is working successfully.

7. **<[?:Image Page URL=**Enter Image url **][==:Image Page URL]>** This is a 'belt and braces' to make sure that the user has entered something to update, with the Alt+click on the Text template field name. If it still contains the prompt '**Enter Image url **', it will reprompt for input.

8. **<[?:var.Set Event Flags=** Set Event Flags y/n **][==:var.Set Event Flags]>** As we could not see the text template on the ORA Control Panel as it was defined with the prefix var. , we can prompt for content in the Auto Type section. The ORA interpreter makes sure that all fields, etc are present in the whole Auto Type template, before it will output to our target application. I test for a specific value of y(upper or lower case ok) and if it is true, ie the y, then I process the following field or library element content.

9. **<[?:var.Update Fact Data=** Update Fact Date and Place y/n **][==:var.Update Fact Data]>** Similar functionality as per 8 above.

10. **[lib.InitVars]** The first call to a library template in this auto type template, and it currently has 93 rows, and will grow as more variants of fieldnames appear! It is an attempt to consolidate the various field names for the same content across the different Services. By convention, I decided to name my consolidated result field name with a leading z. For example, Ancestry refers to the Source title as [Source.Title]; Findmypast refers to the Source title as [Record Set]; and Family Search refers to it as [Source.Collection]. I consolidate all three of these into one [zSource], which I can then use throughout an Auto Type template, irrespective of which Service I am using. This call to load the Initialise Variables library template, [lib.InitVars] ,**MUST** be made before any usage of a z*FieldName* in the AutoType template.

11. **<[?:var.Set Event Flags=y][lib.SetFlag:zEventType] |[lib.SetFocusonEvent:zEventType]>** This is an **if..then..else..** statement enclosed by the < and > chevrons which contain the statement operation to only those lines of code or scripting. This line of the autotype gives me the option of whether I want to update my Flags Tab on the Property box page, or not. The **if** part [?:Set Event Flags=y] explicity tests that the Set Event Flags contains a y, in either upper or lower case. If it does, then it executes the **then** part, the lib.SetFlag library element with the parameter zEventType, and when that completes, it ignores whatever comes after the **|** (pipe character which means Or) and will continue the AutoType with the next line of template after

the closing >. If the [?:Set Event Flags=y] fails, ie it does not contain the y, it will execute the **else** part, ie the code that comes after the |, the lib.SetFocusonEvent library element with the parameter zEventType up to the closing > The process flow will then continue with the next line of the AutoType template. Either way, the Family Historian cursor will be returned to the correct place on the Property Box Main page, as the lib.SetFlag includes the lib.SetFocusonEvent library template as its last operation.

12. **<[?:var.Update Fact Data=y][lib.WriteFactData:zEventType]>** Similar to line 11, this is an explicit test for a y response to the Update Fact data prompt we encountered on line 9 above. If the UpdateFactData contains a y, then execute the lib.WriteFactData library element with the parameter zEventType, else do nothing. The SetFocusonEvent library template had positioned us in the correct field. This is important for the next steps of writing the Source and Citation will cite against the last located entry in the Property Box.

13. **[lib.WriteSource:zSource]** This library template moves the cursor to open the Source and Citation pane and Selects the Cite Existing Source option, and filters for the Source with the value of the zSource parameter. As ORA does not know the content of your list of Sources, it is completely up to you, the user, to ensure that the zSource title has been created prior to running the Autotype template. This library template could be modified to Add a New Source from Template, or Add a New Generic Source. As a source 'Lumper' I use generic Sources using the zSource title gleaned from the Service website, but a 'Splitter' would have to create their own zSource title according to their particular source file naming convention. I would do this in a Text Template where I could construct the Source title, and use that field as the parameter. It would also possibly mean replacing my [tab*8} etc with commands you will need to populate your Source Record. See *My Source/Citation layout* later in this document. The {tab*8} skips over all the fields that I should have populated when creating the generic source, adds the date, sets the assessment, and adds the website url.

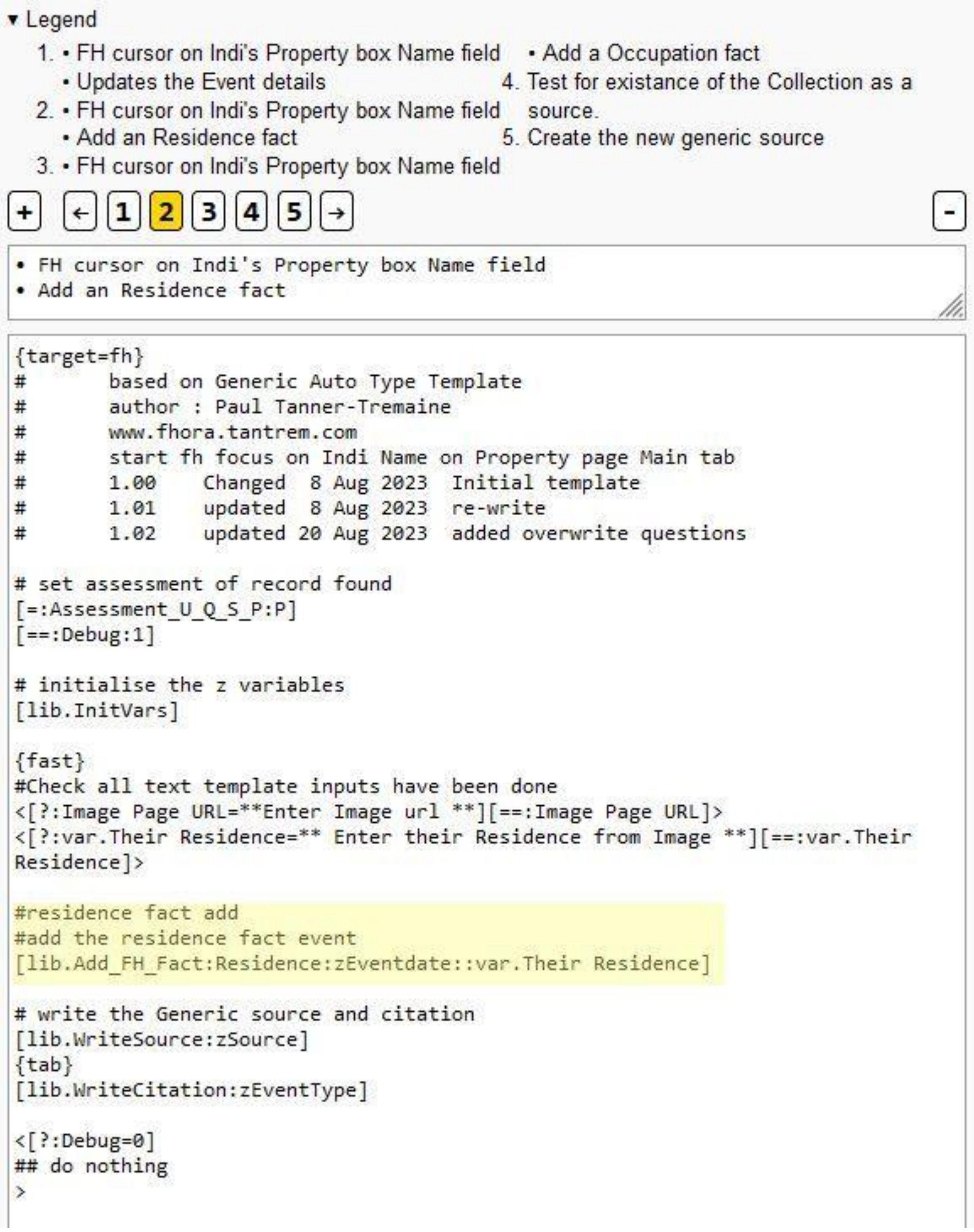
14. **{tab}** This moves the cursor from the Source section to the Citation Text from Source section.

15. **[lib.WriteCitation:zEventType]** This writes the Text from Source Citation data in my standard free-form layout, and can be completely modified the change the

field you want to capture, or the sequence of the fields.

16. **<[?:Debug=0] ## do nothing >** This is the ‘other’ part of my Debug racing methods, and will have no adverse effect on the final running of the Auto Type template.

Template button 2



▾ Legend

1. • FH cursor on Indi's Property box Name field
   • Updates the Event details
2. • FH cursor on Indi's Property box Name field
   • Add an Residence fact
3. • FH cursor on Indi's Property box Name field
   • Add a Occupation fact
4. Test for existance of the Collection as a source.
5. Create the new generic source

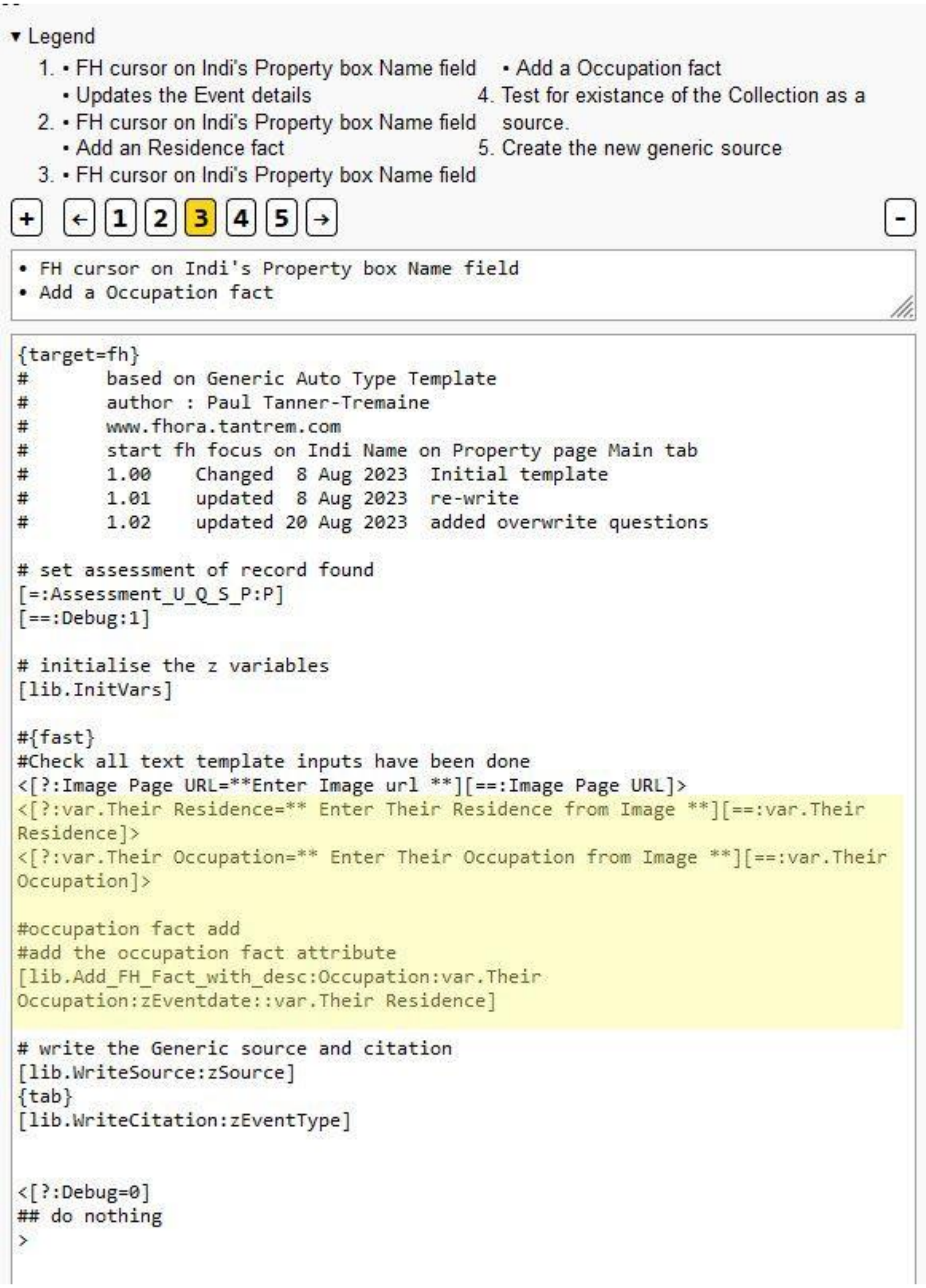
```
• FH cursor on Indi's Property box Name field
• Add an Residence fact
```

```
{target=fh}
#       based on Generic Auto Type Template
#       author : Paul Tanner-Tremaine
#       www.fhora.tantrem.com
#       start fh focus on Indi Name on Property page Main tab
#       1.00    Changed  8 Aug 2023  Initial template
#       1.01    updated  8 Aug 2023  re-write
#       1.02    updated 20 Aug 2023  added overwrite questions

# set assessment of record found
[=:Assessment_U_Q_S_P:P]
[==:Debug:1]

# initialise the z variables
[lib.InitVars]

{fast}
#Check all text template inputs have been done
<[?:Image Page URL=**Enter Image url **][==:Image Page URL]>
<[?:var.Their Residence=** Enter their Residence from Image **][==:var.Their
Residence]>

#residence fact add
#add the residence fact event
[lib.Add_FH_Fact:Residence:zEventdate::var.Their Residence]

# write the Generic source and citation
[lib.WriteSource:zSource]
{tab}
[lib.WriteCitation:zEventType]

<[?:Debug=0]
## do nothing
>
```

The only new point to note in this Auto Type button, is highlighted. Note also the prompt line above it to capture His residence.

To add the residence, I use the Add_FH_Fact library template, which in it's generic form, has 4 parameters passed to it.: factname:date:street address: place. As we don't have the street address of his residence, we can leave it out, and the library template will ignore it in the output. See the *library template* discussion below.

Template button 3



▾ Legend

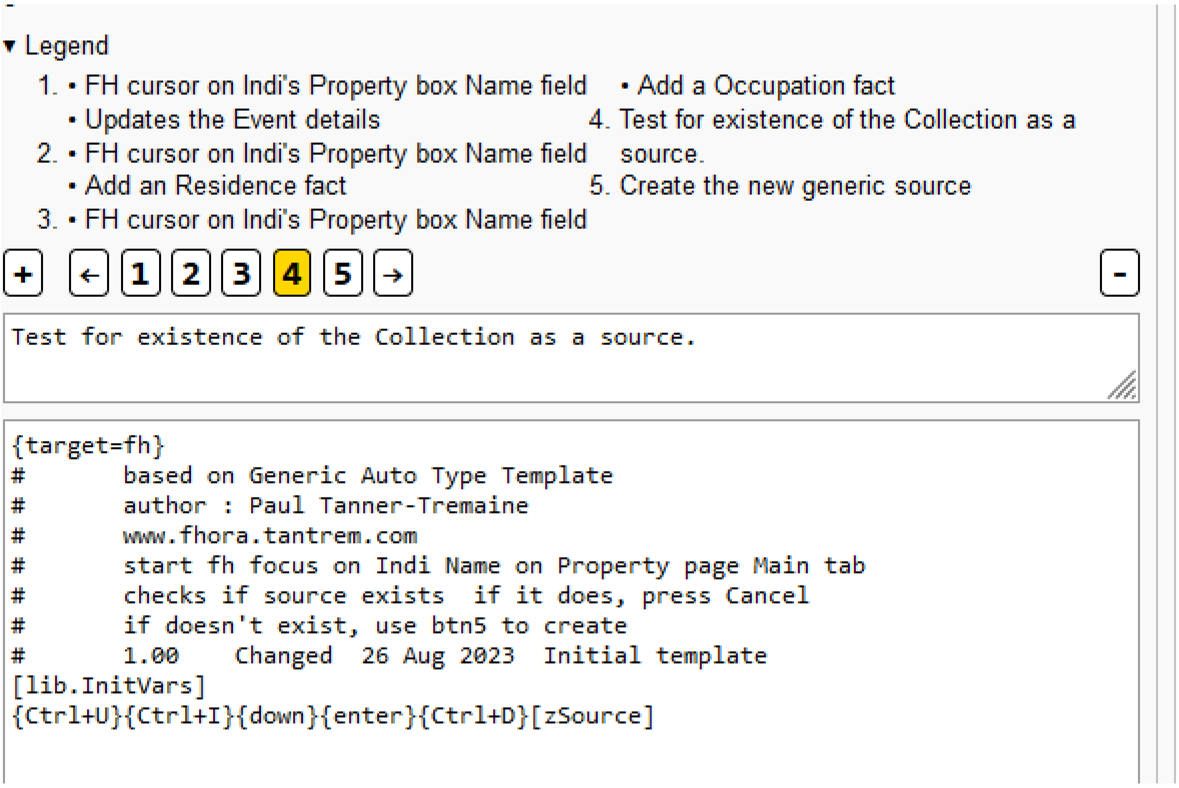
1. • FH cursor on Indi's Property box Name field
   • Updates the Event details
2. • FH cursor on Indi's Property box Name field
   • Add an Residence fact
3. • FH cursor on Indi's Property box Name field
   • Add a Occupation fact
4. Test for existance of the Collection as a source.
5. Create the new generic source

```
• FH cursor on Indi's Property box Name field
• Add a Occupation fact
```

```
{target=fh}
#       based on Generic Auto Type Template
#       author : Paul Tanner-Tremaine
#       www.fhora.tantrem.com
#       start fh focus on Indi Name on Property page Main tab
#       1.00    Changed  8 Aug 2023  Initial template
#       1.01    updated  8 Aug 2023  re-write
#       1.02    updated 20 Aug 2023  added overwrite questions

# set assessment of record found
[=:Assessment_U_Q_S_P:P]
[==:Debug:1]

# initialise the z variables
[lib.InitVars]

#{fast}
#Check all text template inputs have been done
<[?:Image Page URL=**Enter Image url **][==:Image Page URL]>
<[?:var.Their Residence=** Enter Their Residence from Image **][==:var.Their Residence]>
<[?:var.Their Occupation=** Enter Their Occupation from Image **][==:var.Their Occupation]>

#occupation fact add
#add the occupation fact attribute
[lib.Add_FH_Fact_with_desc:Occupation:var.Their Occupation:zEventdate::var.Their Residence]

# write the Generic source and citation
[lib.WriteSource:zSource]
{tab}
[lib.WriteCitation:zEventType]


<[?:Debug=0]
## do nothing
>
```

Again, prompts to capture to capture his Occupation and Residence, and a call to the library element Add_FH_Fact_with_desc with 5 parameters; fact type, the fact attribute description of his occupation, the date, street address and place. Note that the 4th parameter here is also blank.

Template Button 4



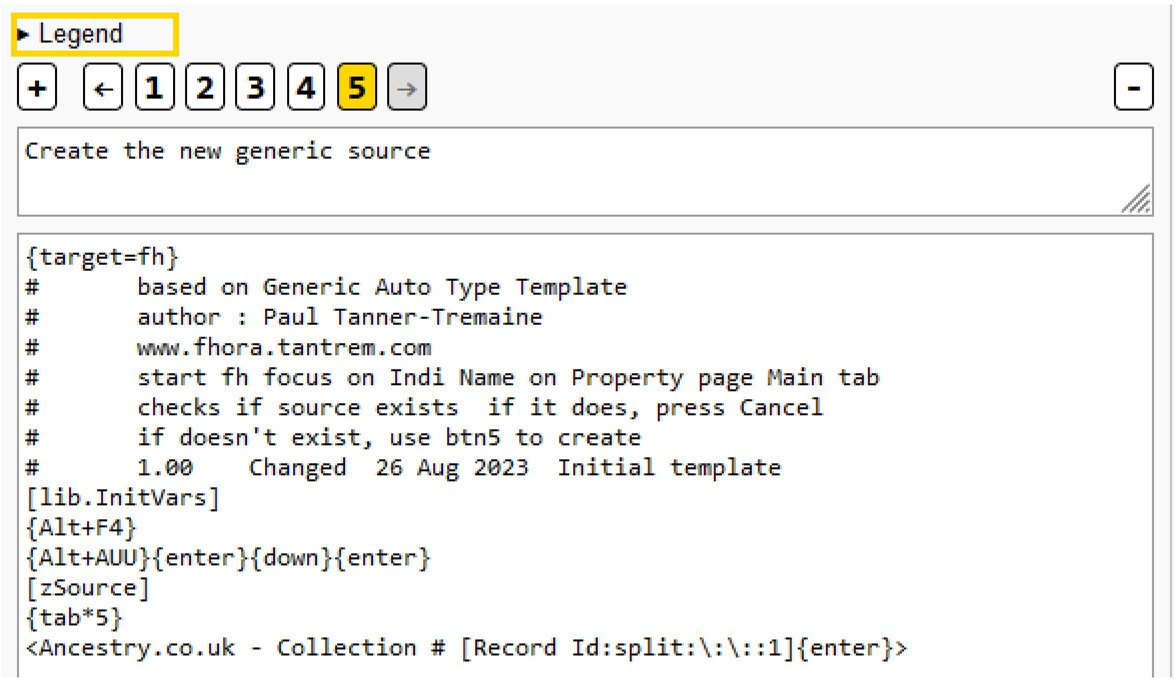
1. • FH cursor on Indi's Property box Name field
   • Updates the Event details
2. • FH cursor on Indi's Property box Name field
   • Add an Residence fact
3. • FH cursor on Indi's Property box Name field
   • Add a Occupation fact
4. Test for existence of the Collection as a source.
5. Create the new generic source

```
Test for existence of the Collection as a source.
```

```
{target=fh}
#       based on Generic Auto Type Template
#       author : Paul Tanner-Tremaine
#       www.fhora.tantrem.com
#       start fh focus on Indi Name on Property page Main tab
#       checks if source exists  if it does, press Cancel
#       if doesn't exist, use btn5 to create
#       1.00    Changed  26 Aug 2023  Initial template
[lib.InitVars]
{Ctrl+U}{Ctrl+I}{down}{enter}{Ctrl+D}[zSource]
```

One of the issues I had when testing, was that I did not already have the 'new' Source in Family Historian, and my template would run amok. This Button 4 just runs to see if the Source is available or not. Click the Cancel button either result. If you don't have the Source already in your Sources list, Button 5 below should create it for you.

Template Button 5



```
Create the new generic source
```

```
{target=fh}
#       based on Generic Auto Type Template
#       author : Paul Tanner-Tremaine
#       www.fhora.tantrem.com
#       start fh focus on Indi Name on Property page Main tab
#       checks if source exists  if it does, press Cancel
#       if doesn't exist, use btn5 to create
#       1.00    Changed  26 Aug 2023  Initial template
[lib.InitVars]
{Alt+F4}
{Alt+AUU}{enter}{down}{enter}
[zSource]
{tab*5}
<Ancestry.co.uk - Collection # [Record Id:split:\:\::1]{enter}>
```

The Library Templates

These are the Library templates that are used in the Text Templates and the AutoType templates called or used in this Generic BMD Autotype template. I would strongly recommend that you examine the actual Library Template in your browser as you read through this section. Some of the Library templates are rather long, and would be difficult to add in their entirety at this point. The actual library element code is available from my fhora website Template Library Report

[lib.DividerBar]

This creates a divider bar in the Control Panel. I use it as the last text template in a Service Collection, so that my other 'input' text templates appear above it, until they have had an input or refresh.

[lib.SetAssesment]

This library element uses the preset field [=:Assessment_U_Q_S_P:P] to test for the parameter, and then outputs the required number of {down}s to select the appropriate value for the record assessment. For some reason, ORA requires the actual number of downs plus 1 to select the value, in most dropdowns – be careful.

[lib.InitVars]

ORA does a fantastic job of scraping websites and producing the fields and the values of those fields. Unfortunately, the different Services that I commonly use; Ancestry, FindMyPast and FamilySearch, do not or have not standardised on any field naming conventions, even among their own Collections! Sometimes a Baptism place is known as Baptism Place, or Christening Place, or Event Place, or Event Place (original) or just Place. I decided that I would try and create a list of fields beginning with a z, ie zBaptismplace, common between the Services, and populated with the data assigned in any of their variants. ORA does know which Service we are using and we can test that, and branch to any Service's set of *zFieldnames*. Therefore, in Auto Types, after I have called/run the [lib.InitVars] library element, I can refer to a *zFieldname* and and not worry about all the different names that ORA initially finds. We can also use this approach to determine the EventType that the Collection comes from, and test that result as well or pass it as a parameter to a library template. If I decide to add another Service to my current three, ie MyHeritage or The Genealogist, I can just add a new section in this library, and hopefully cover all their variants as well, without having to rewrite my Auto Type templates.

[lib.SetFlag:zEventType]

In my Family Historian application, I have added a Flags tab in the Property Box, so I have the tabs Main, Facts, Tabs, Census UK, Notes, Media and All. See *My Flags Tab Layout*
To navigate to the Flags tab from the Main tab requires two {Ctrl+tab} keyboard shortcuts. Each of the Flags can be toggled on or off with a spacebar press or mouse click. Depending on the zEventType that is passed through, the relevant number of {tab}s are applied before the {space} to set the flag.

Once the Flags have been set, we navigate back to the Main Tab on the Property Box, and call the library template [lib.SetFocusonEvent:param] to position the cursor at the right zEventdate field ready for the next input.

[lib.SetFocusonEvent:zEventType]

This library template moves the cursor from the INDI name input field on the Property Box, to the relevant Date field depending on the value of the parameter zEventType. From there, we can get ORA to either write the date and place info, or skip over the existing info, and open up the Source and Citation for that zEventType.

[lib.WriteFactData:zEventType]

Depending on the value of the parameter, zEventType, the relevant Date and Place fields will be output to Family Historian, using the z*Fieldnames* initialised in the InitVars library template.

[lib.WriteSource:zSource]

This is a simple template plays an important part in my citing of Sources for an event or attribute. The {250} and {500} are quarter of a second, and half a second delays for the ORA output, so that any background processing in Family Historian has a chance to catch up. The {Ctrl+U}{Ctrl+I}{down}{enter}{Ctrl+D} string of commands are the equivalent of manually opening up the Source/Citation ‘yellow’ pane, either docked below or to the right of the Property Box {Ctrl+U} , the {Ctrl+I} sets it into the Insert Mode, the popup at the bottom of the source pane, the {down} moves the focus point down once to the Cite Existing Source item, and the {enter} selects it and opens up the Source Selection pop-up window. The {Ctrl+D} moves the cursor to the Filter input box. The [{1}] tells ORA to output the content of the first parameter, which when we initiated the call to this Library template was zSource, which would have contained “Somerset, England, Marriage Registers, Bonds and Allegations, 1754-1914” or another user constructed Source Title field. As this filter field gets populated, Family Historian starts the filtering process, matching alphabetically, and eventually should find the only matching source which we had previously Added to the Family Historian Source list. The {F6} moves the cursor to the first record in the results list and outlines it with a dotted line. The {space} is equivalent of highlight that result with the keyboard space bar, not of entering a space! The {enter} now selects that highlighted row and opens up the Source and Citation window, with the Title set, and any other Source detail that you may have entered when creating the Source record. If you change ANY of the shown fields in this section now, those changes will be applied to every instance where this Source is referenced. I skip over these fields with the {tab*8} ie tab 8 times, which gets the focus point down to the Entry date. The field [Page.Access date] will always have the current date, so we can output that into the Entry Date field. Now we {tab} to the assessment field where we call the [lib.SetAssessment] library template. In this autotype template we originally set [=:Assessment_U_Q_S_P:P] and the library template will associate the parameter P with the right number of down arrow keystrokes to accept the Primary vale for the drop down, and enter that in the field. Once that has been down, we {tab} to the Where Within: field, and it gets populated with the content of the scraped [URL] field. This gives me a 'bread-crumb' trail back to where I originally found the Source record.

[lib.WriteCitation:zEventType]

This Library Template is probably one of the most complex I have built and difficult to describe. It is reliant on just one parameter, the zEventType, and an evaluation of which Service we are utilising. I have also identified what data I would like to have as Common Data per Service, and Event Specific data per zEventType. In the template the zEventType is represented by the <[?{1}=] In simple logic terms we have a lot of if..then..else type expressions, eg

```
< If the Service=Ancestry  ( testing the value of o.Service )
        Then output the Common Data applicable to Ancestry.
        < If the parameter [{1}] contains Birth
                Output the specific data applicable to a Birth record
        >|  or
        <if the parameter [{1}] contains Baptism
                Output the specific data applicable to a Baptism record
        >| or

        <if the parameter [{1}] contains Marriage
                Output the specific data applicable to a Marriage record
        >| or
ditto for Death, Burial, etc
        >
>   the end of the Ancestry service
```

Now we test for the next Service we have, ie Findmypast

```
< If the Service=Findmypast  ( testing the value of o.Service )
        Then output the Common Data applicable to Findmypast.
        < If the parameter [{1}] contains Birth
                Output the specific data applicable to a Birth record
        >|  or
        ditto for all the zEventType parameter content until
        > the end of the zEventType tests
>    the end of the Findmypast service
```

and duplicated structure for any other Services we want to process in this common library template.

Within each Specific data area in the zEventType by Service Type, optional fields are enclosed with their own set of chevrons < > and will not be output if the field is empty or does not exists.

After we have got through all these tests by Service type, you will note I open the Rich Text Editor with the {Ctrl+space} and allow a {500} half second delay for FH to catch up.

The cursor is moved to the end of the current text with (Ctrl+End} and {enter} to begin a new line before the test to see if the Image Page URL has content which is not the original prompt. If this is true, process the library template [lib.AddImageLinktoTFS] which takes the two parameters of the Image Page URL and the descriptive text by using the Insert Web Link commands, followed by the similar library template that inserts the Web Link for the record URL and descriptive text. Finally we close the Write Citation and the Source Citation pane with the {Alt+F4}{enter}.

[lib.AddImageLinktoTFS:Image Page URL:zImagehrefdesc]

A very simple library template that only works when you are writing to a Rich Text Editor enable field. The {SHIFT+F10}{250}{UP*2}{RIGHT}{DOWN}{ENTER}{250} string of commands opens up the Add Link popup withing the editor, selects the Web link and then completes the Web Address which is within the first parameter[{1}] and then the Display text from the second parameter[{2}]

[lib.AddRecordLinktoTFS:Page.URL:zURLhrefdesc]

An almost identical library template to the previous one as it has the same logic, but differently named parameters. I kept these two library templates with different names so I did not get confused with which I was loading at that step in the autotype.

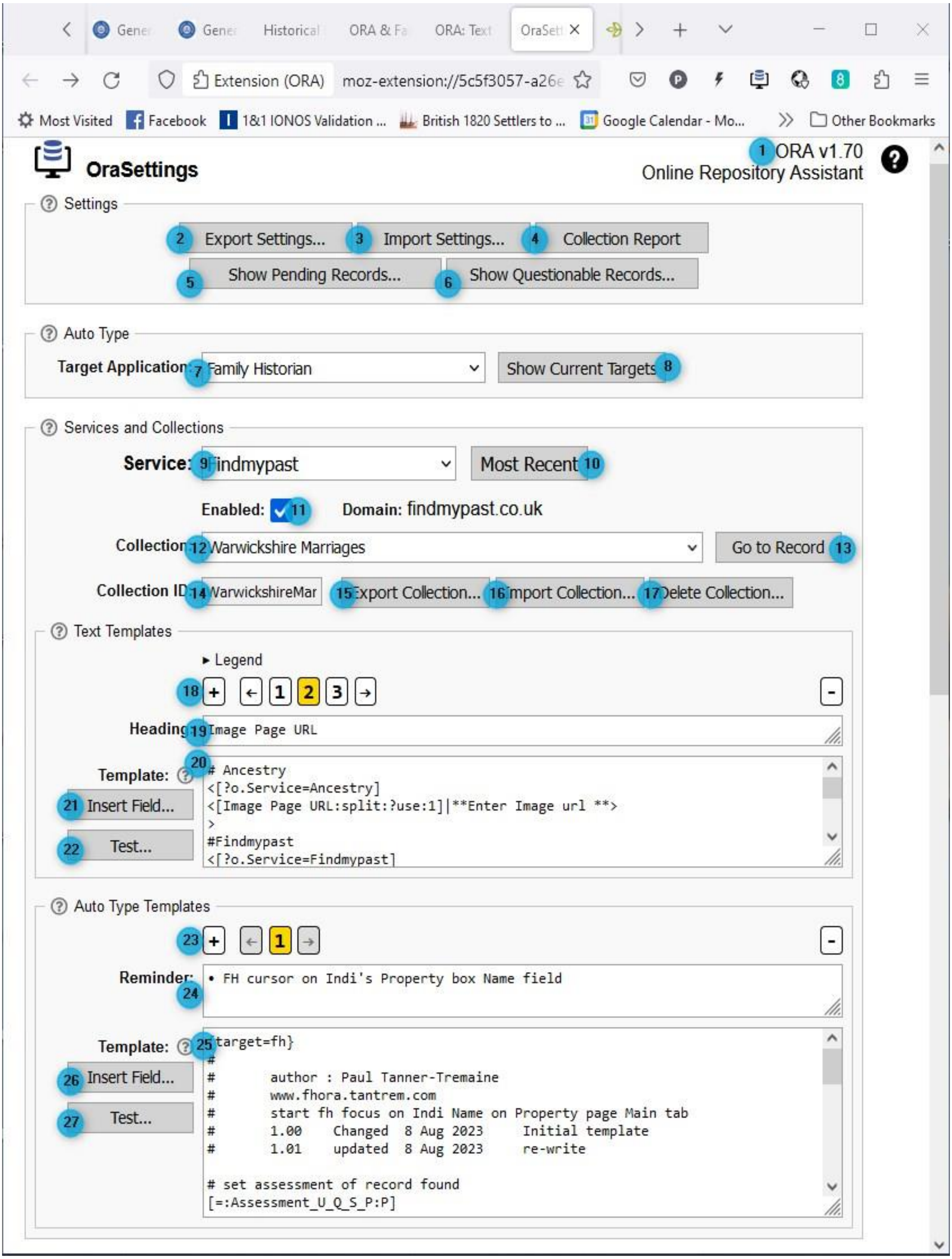
[lib.Add_FH_Fact:Residence:zMarriagedate::var.His Residence]

This template is designed to create and write out the Date, Age and Place of fact EVENT type record. It has 4 parameters passed to it. : factname:date:street address:place. The template first moves to the Fact Tab of the property box with the {Ctrl+tab} command, and then set in the Insert mode with the {Ctrl+I} moving to the filter input field. The first parameter is used as a literal value with the {1}, in this case the Residence event, and is selected with the {enter}. It then {tab}s to the Date field and post the content of the second parameter [{2}], {tab} to the Age field and enter the persons Age if it is available, before {tab}ing to the Place field where it will optionally enter the third [{3}] if we have the street address, and finally the His.Residence from [{4}]. The last {tab} moves the cursor on a step so that the data is actually written away to Family Historian.

[lib.Add_FH_Fact_with_desc:Occupation:var.His Occupation:zMarriagedate::var.His Residence]

This Library template is almost a replica of the one above, except that it caters for fact ATTRIBUTE record addition which has a description field, so five parameters as factname:fact value:date:street address:place. This allows us to enter attribute type facts such as Occupation, Labourer, date of event, street address, and Place.

The ORA Settings page



Target Application: Family Historian

Service: findmypast

Enabled: ☑ Domain: findmypast.co.uk

Collection: Warwickshire Marriages

Collection ID: WarwickshireMar

Heading: Image Page URL

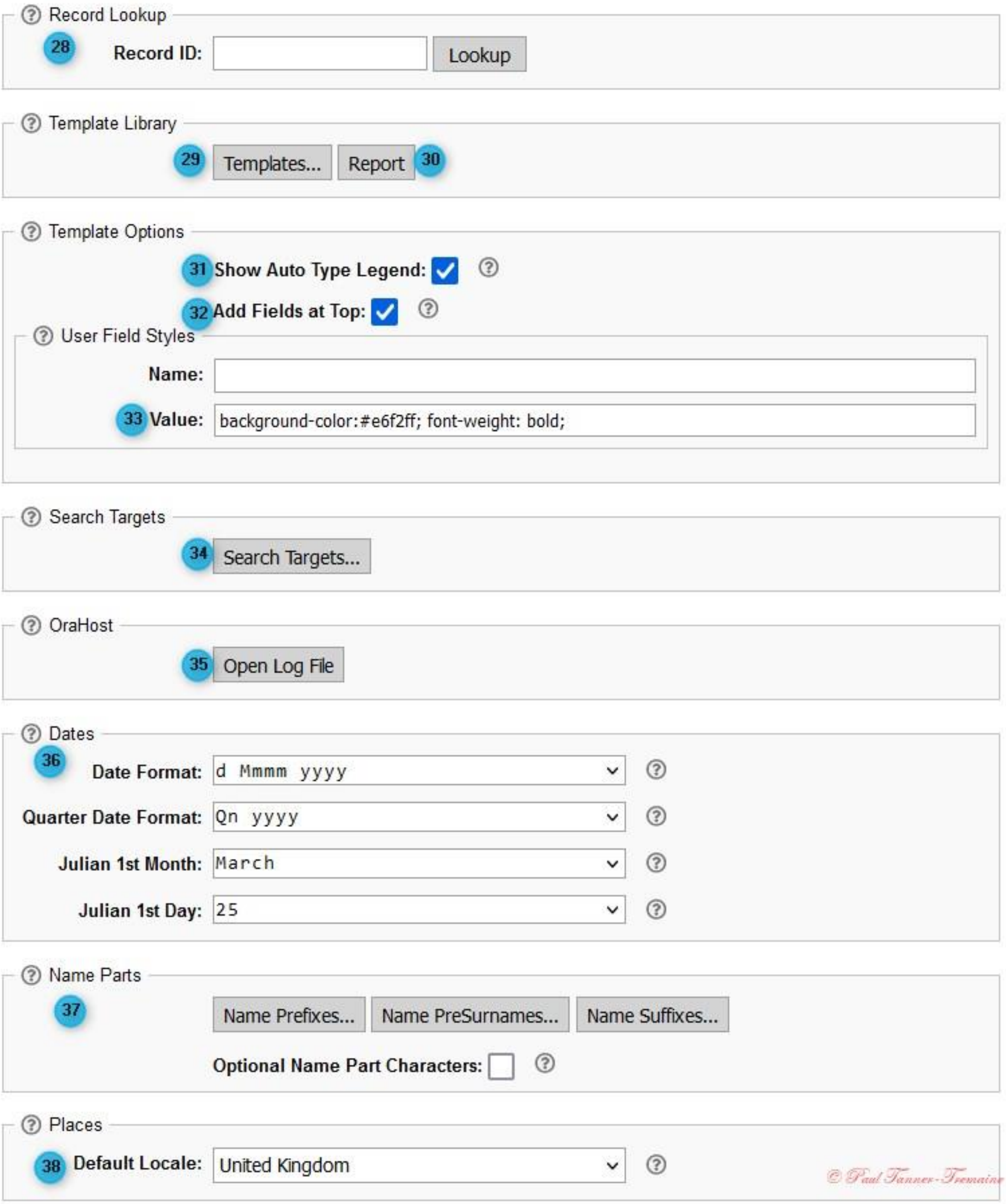
Template:

```
# Ancestry
<[?o.Service=Ancestry]
<[Image Page URL:split:?use:1]|**Enter Image url **>
>
#Findmypast
<[?o.Service=Findmypast]
```

Reminder: • FH cursor on Indi's Property box Name field

Template:

```
target=fh}
#
#       author : Paul Tanner-Tremaine
#       www.fhora.tantrem.com
#       start fh focus on Indi Name on Property page Main tab
#       1.00    Changed  8 Aug 2023     Initial template
#       1.01    updated  8 Aug 2023     re-write

# set assessment of record found
[=:Assessment_U_Q_S_P:P]
```



Record Lookup

28 Record ID: [ ] Lookup

Template Library

29 Templates... Report 30

Template Options

31 Show Auto Type Legend: ☑

32 Add Fields at Top: ☑

User Field Styles

Name: [ ]

33 Value: background-color:#e6f2ff; font-weight: bold;

Search Targets

34 Search Targets...

OraHost

35 Open Log File

Dates

36 Date Format: d Mmmm yyyy

Quarter Date Format: Qn yyyy

Julian 1st Month: March

Julian 1st Day: 25

Name Parts

37 Name Prefixes... Name PreSurnames... Name Suffixes...

Optional Name Part Characters: □

Places

38 Default Locale: United Kingdom

All these blue Bullet points are described on the following pages:

The ORA Settings page Field Meanings

The following refers to the blue numbered buttons on the previous 2 pages

1. This displays the **current version** of your ORA Extension. If you hover your mouse over the version number, ie the 1.70, a pop-up will show the ORA Host version. Clicking on the black Question Mark will take you to The ORA Help pages, where the Contents tab is most useful.
2. The **Export Settings** button Link will export ALL of your development at the current time to a file in your Documents\ORA\Settings folder. Use this as a Backup mechanism in tandem with the Import Settings.
3. The **Import Settings** button Link will restore a previously saved exported settings file from your Documents\ORA\Settings folder, overwriting your current content.
4. The **Collection Report** button Link will generate an HTML web page of all your Autotypes, sorted by Service and Collection and will display on your browser tabs.
5. **Show Pending Records** will list all those records that you have assigned a P, or Pending status as you have viewed them in the browser during your research.
6. **Show Questionable Records** will list all those records where you have assigned a ? question mark or Questionable Status
7. This drop-down list will show all the available **Target Applications** that you can use in ORA (we should be using Family Historian!)
8. **Show Current Targets** will show you which applications you have running at the present time. If you have 2 or more instances of Family Historian running, close them down and only leave one running, otherwise ORA will not know where to send the output from an autotype template!
9. The **Service** dropdown window will show the Service that was being displayed by your web browser when you last ran or refreshed an Autotype template. The dropdown will show a list of the **Current Services**(web sites) that are available to you.
10. **Most Recent** will change the Service and Collection to the most recent. Yes, I know it gets confusing, and one day I will sort it all out in my own mind!😊
11. The **Enabled** tick box, if ticked, allows ORA to display its Control Panel on that domain's websites.
12. This is the **Collection** Name that is added to your set of ORA Settings when you visit a new Collection displayed by a Service. It is typically the Source.Title that is used as an identifier by the Service provider, and in many cases forms the Source name for our source in Family Historian. If it is the first time you have visited that particular Collection, the Text Template and Auto Type Template regions will be blank. This

indicates that you have not yet created an Autotype or Text template for that collection provided by that Service. The dropdown will show you ALL the Collections that you have visited. You can select any Collection that is in your system to view the Templates you may have created.

13. **Go to Record**. I regularly use this button, particularly in my testing of templates and library templates. If you do not have that ‘Service/Collection’ currently open in one of your browser tabs, it will go and open a new tab with that ‘last used’ record for you.
14. **Collection ID**. This is a value created by the Service provider, most likely as an Index to their collections.
15. **Export Collection**. This will allow you to export that set of Service Collection Text Templates and Auto Type templates that are currently being displayed, either to the clipboard or to a file.
16. **Import Collection.** This button allows the import of a set of Service Collection Text and Auto Type templates, either from the clipboard or from file. The combination of using this Export and Import allows you to effectively copy and paste an AutoType and Text template from one Collection to another, where you may make minor adjustments to suit that new Collection, rather than re-typing it all!
17. **Delete Collection** will do what it says on the tin – it will delete that set of text and autotype templates from your system. Be careful! The only way to get it back would be to Import the collection if you had previously exported it to file- points 15&16, or to Import Settings, which will overwrite everything back to the last time you Exported the Settings – points 2&3 above
18. **The Add +, Move <- -> ,Delete –** icons that if clicked will add, move or delete a Text Template, which will be given a number for that Text Template. The ‘active’ or shown template’s number will be yellow highlighted. If you click the little black arrow next to ▸ Legend just above the row of icons, it will toggle the display of the Headings of the templates.
19. **Heading:** This is the input field for the Heading or title of your text template. Typically, this will be the field name for the text template.
20. **Template:** This is the input area where you will write the ORA expressions or code for the text template.
21. **Insert Field…** Clicking this will open the Insert Field Reference pop-up window, allowing you to select one of the fields that ORA has scraped from the underlying html code for the current active Service Collection, and some ‘hidden’ Page.fields which are very useful in the scripting or coding of the templates.
    a. The selection dropdown area. Click on a field that you wish to insert at your cursor focus point in the Template area.

b. Insert. This will insert the fieldname with its standard brackets, ie [fieldname], and when the template is run or tested, will output the content of the field.

c. Insert (conditional). This will insert the fieldname inside single chevrons, ie <, [fieldname]>. The chevrons indicate that the content will only be output if the [fieldname] field has a value. The comma and space can be removed if necessary. They are there as a help when creating a Place name from more than one [field]. For example <[Church]><, [Town]><, [County]><, [Country]> could result in St Pauls, Hoxton, Middlesex, England if all fields had data. If the Town field had no data it would produce St Pauls, Middlesex, England.

d. Insert (conditional with name). Similar to point c above but includes the fieldname as well as the field. For example <, First name(s): [First name(s)]> Again, useful when creating citations or notes. Personally I remove the leading comma and space as I normally have these on a new line.

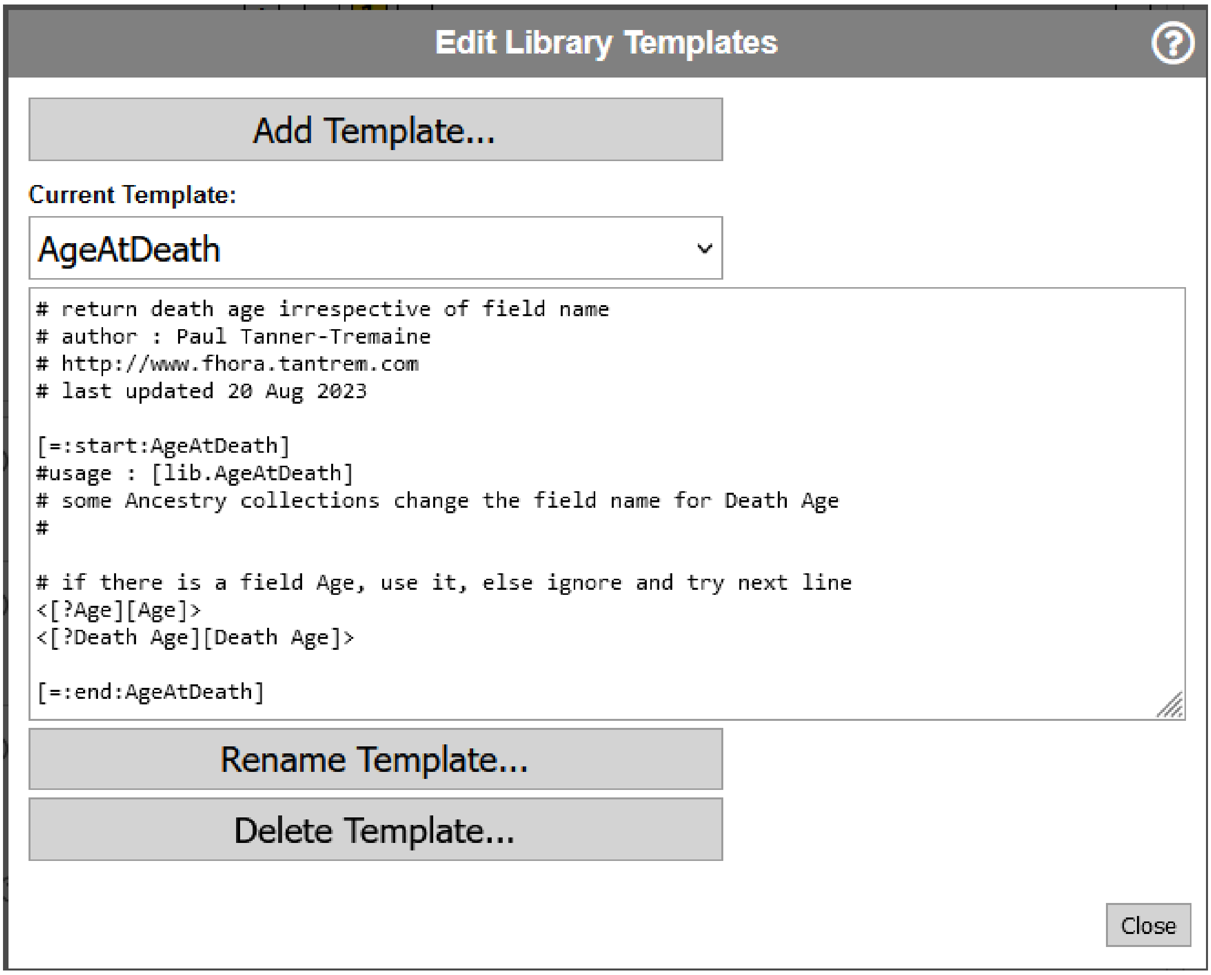
e. Insert (name only). Inserts just the field name.

This Input Field Reference helps to avoid having typos in your scripts!

22. **Test.** This button allows you to test the output of the template before activating or running the template or autotype template. If at any time parts of the template **Output** show up bold highlighted in yellow, it means that you have a typo error, or have a mismatched pair of chevrons, brackets or curly brackets. These need to be resolved before you try to run the autotype templates. There is also a **Trace** section that will show you step by step what the ORA interpreter is testing before outputting the results. This trace is invaluable in determining if you have got your conditional logic correct.
23. **The Add +, Move <- -> ,Delete –** icons that if clicked will add, move or delete a Text Template, which will be given a number for that Text Template. The ‘active’ or shown template’s number will be yellow highlighted.
24. **Reminder.** This input area is for you to enter a set of instructions for running the autotype, or a reminder of where to place the cursor in Family Historian as the start focus point for the autotype template. It is fairly free form, and will show on the ORA Control Panel as you hover the mouse over the Autotype button number.
25. **Template:** This is the input area where you will write the ORA expressions or code for the Auto Type template. I advise that you always start the template by defining the intended target for the autotype, ie {target=fh} or {target=excel}, etc. a # at the

beginning (first position) in a line tells the ORA interpreter to ignore that line, so is ideal for comments.

26. **See 21 above.**
27. **See 22 above.**
28. **Record Lookup by Record ID.** According to the ORA help, if you enter a valid record ID for the repository associated with the current service, you can click the [Lookup] button to open the record details page for that record ID. I have never got this to work, so cannot give an example.
29. **Templates.** Clicking this button opens up the Edit Library Templates window. It gives the options to Add Template, Rename Template, Delete Template, and display the Current template by means of selection from the Dropdown input selection. The current template can be edited in the display section, and is 'saved' when you click the Close button.



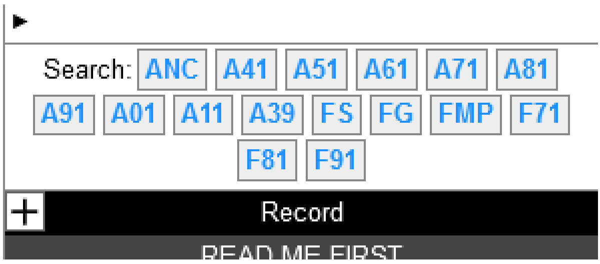
Current Template: AgeAtDeath

```
# return death age irrespective of field name
# author : Paul Tanner-Tremaine
# http://www.fhora.tantrem.com
# last updated 20 Aug 2023

[=:start:AgeAtDeath]
#usage : [lib.AgeAtDeath]
# some Ancestry collections change the field name for Death Age
#

# if there is a field Age, use it, else ignore and try next line
<[?Age][Age]>
<[?Death Age][Death Age]>

[=:end:AgeAtDeath]
```

see further discussion on Library Templates below.

30. **Report** button Link will generate an HTML web page of all your Library Templates, sorted alphabetically by template name.
31. **Show Autotype Legend:** Recommend you leave this ticked!
32. **Add Fields at Top:** If this is ticked, it will display all your Text Template output before the regular fields lists on the ORA Control Panel.

33. **User Field Styles Value**: I have not experimented with this section, other than to change the background-color: value from the default to a pale blue #e6f2ff; which I find easier on the eyes.
34. **Search Targets.** This allows you to edit or add new search targets to the ORA Control Panel. They are quick shortcuts to Services and Collections, with pre-filled data from the current Service and collection you are viewing.

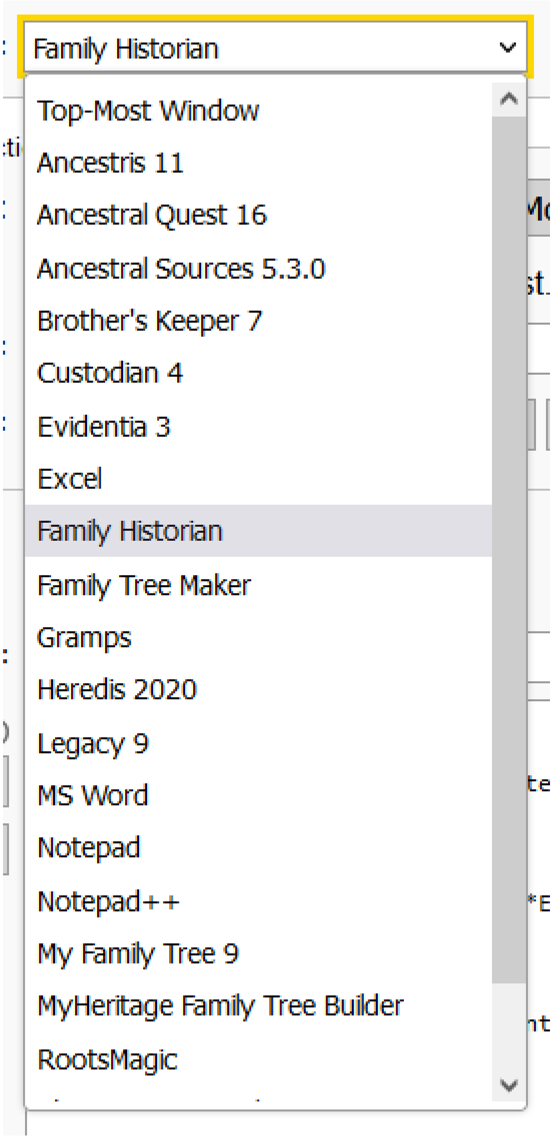

Search: ANC A41 A51 A61 A71 A81
A91 A01 A11 A39 FS FG FMP F71
F81 F91
Record
READ ME FIRST

. See further discussion on Search Targets below,

35. **Open Log File.** The logfile is a trace of the activity of the ORA Settings, and is of use to the author of ORA, John Cardinal if your system goes pear shaped and he requires the log.
36. **Dates.** A section where you can change the Date formats that ORA uses. The defaults are fine for me.
37. **Name Parts**. These are pre-defined prefixes and suffixes to name parts. I have not used these at all.
38. **Default Locale**. A dropdown selection of your country area/region.

Supported Applications as at ORA version 1.70

ORA currently supports the following applications, if they are Enabled.

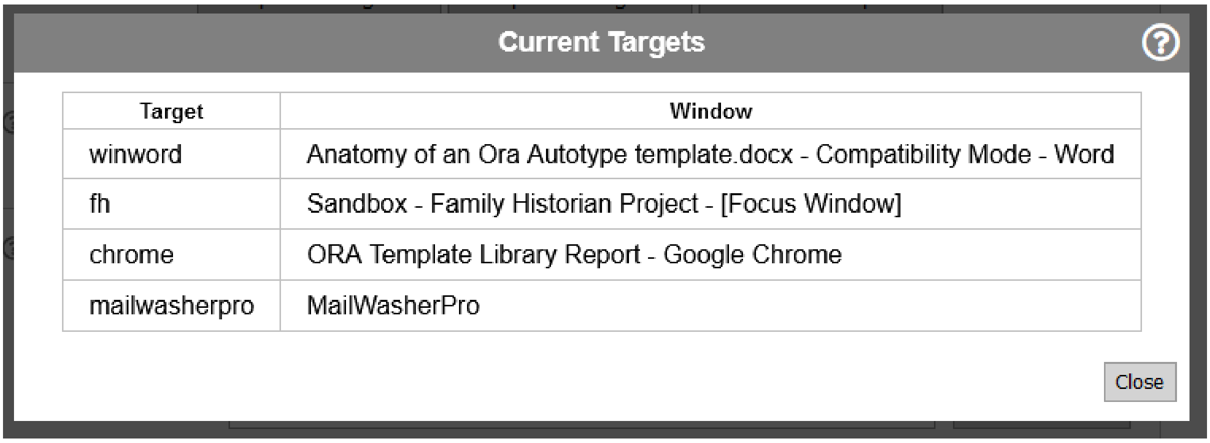


Family Historian

- Top-Most Window
- Ancestris 11
- Ancestral Quest 16
- Ancestral Sources 5.3.0
- Brother's Keeper 7
- Custodian 4
- Evidentia 3
- Excel
- Family Historian
- Family Tree Maker
- Gramps
- Heredis 2020
- Legacy 9
- MS Word
- Notepad
- Notepad++
- My Family Tree 9
- MyHeritage Family Tree Builder
- RootsMagic

Further applications can be requested by emailing John Cardinal, the author, to consider.

Current Targets:



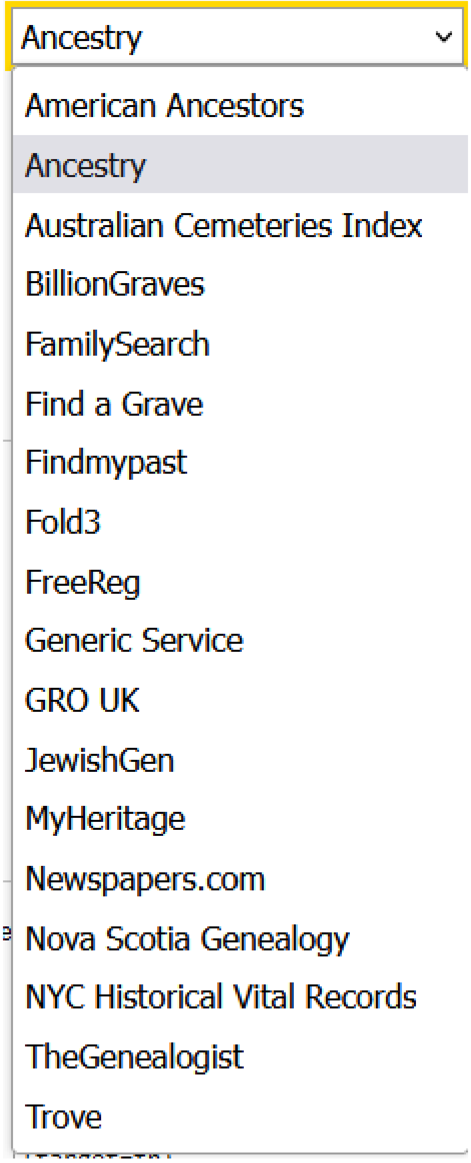
Current Targets

Target	Window
winword	Anatomy of an Ora Autotype template.docx - Compatibility Mode - Word
fh	Sandbox - Family Historian Project - [Focus Window]
chrome	ORA Template Library Report - Google Chrome
mailwasherpro	MailWasherPro

These were the current targets on my system at the time of writing this document, when I clicked on Show Current Targets. *Back to Ora Settings*

Current Services supported in version 1.70

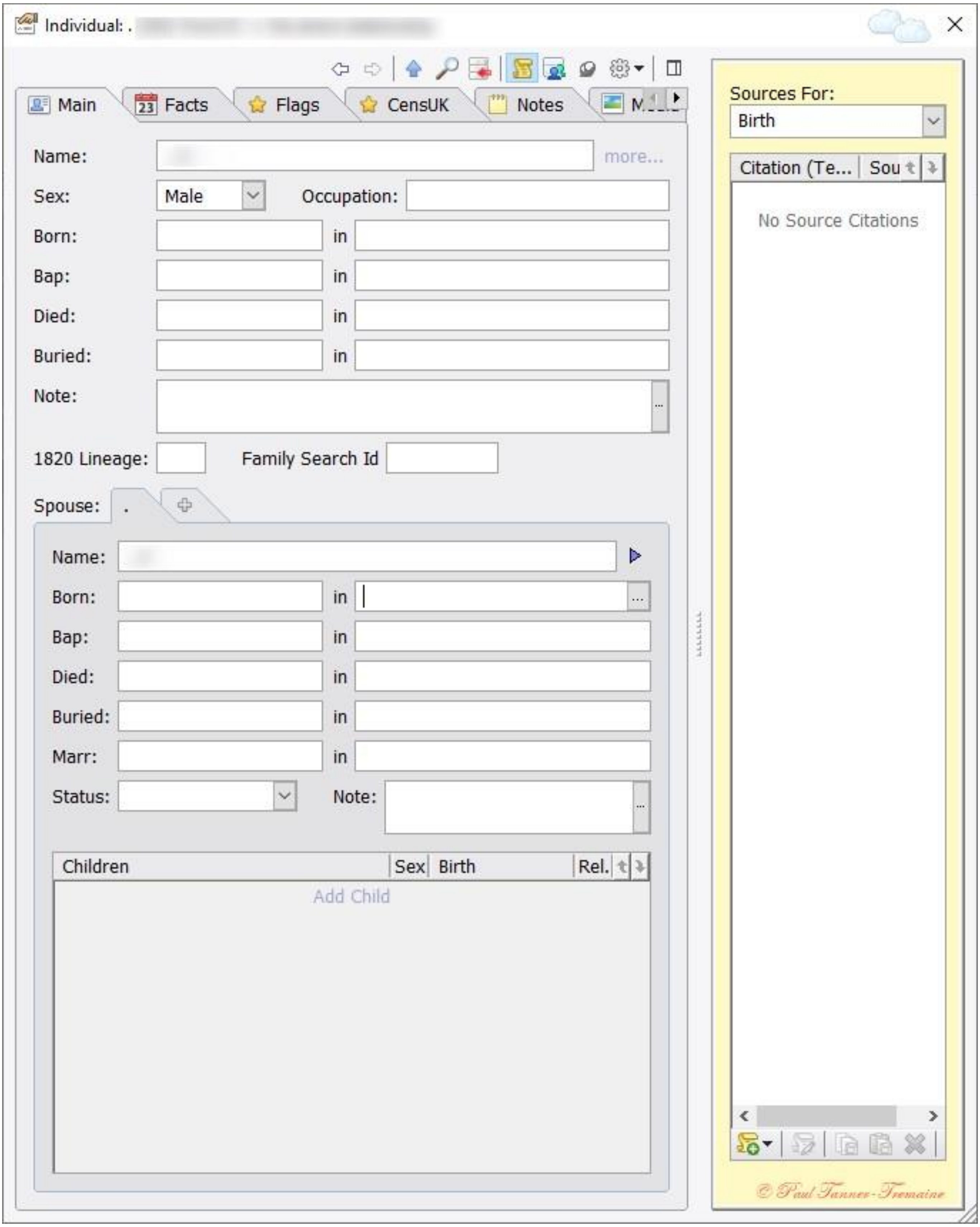
These are the current Websites or Services that ORA will automatically scrape for relevant information, show in the dropdown:



Ancestry

American Ancestors
Ancestry
Australian Cemeteries Index
BillionGraves
FamilySearch
Find a Grave
Findmypast
Fold3
FreeReg
Generic Service
GRO UK
JewishGen
MyHeritage
Newspapers.com
Nova Scotia Genealogy
NYC Historical Vital Records
TheGenealogist
Trove

My Property Box Main Tab layout



Individual: . [redacted]

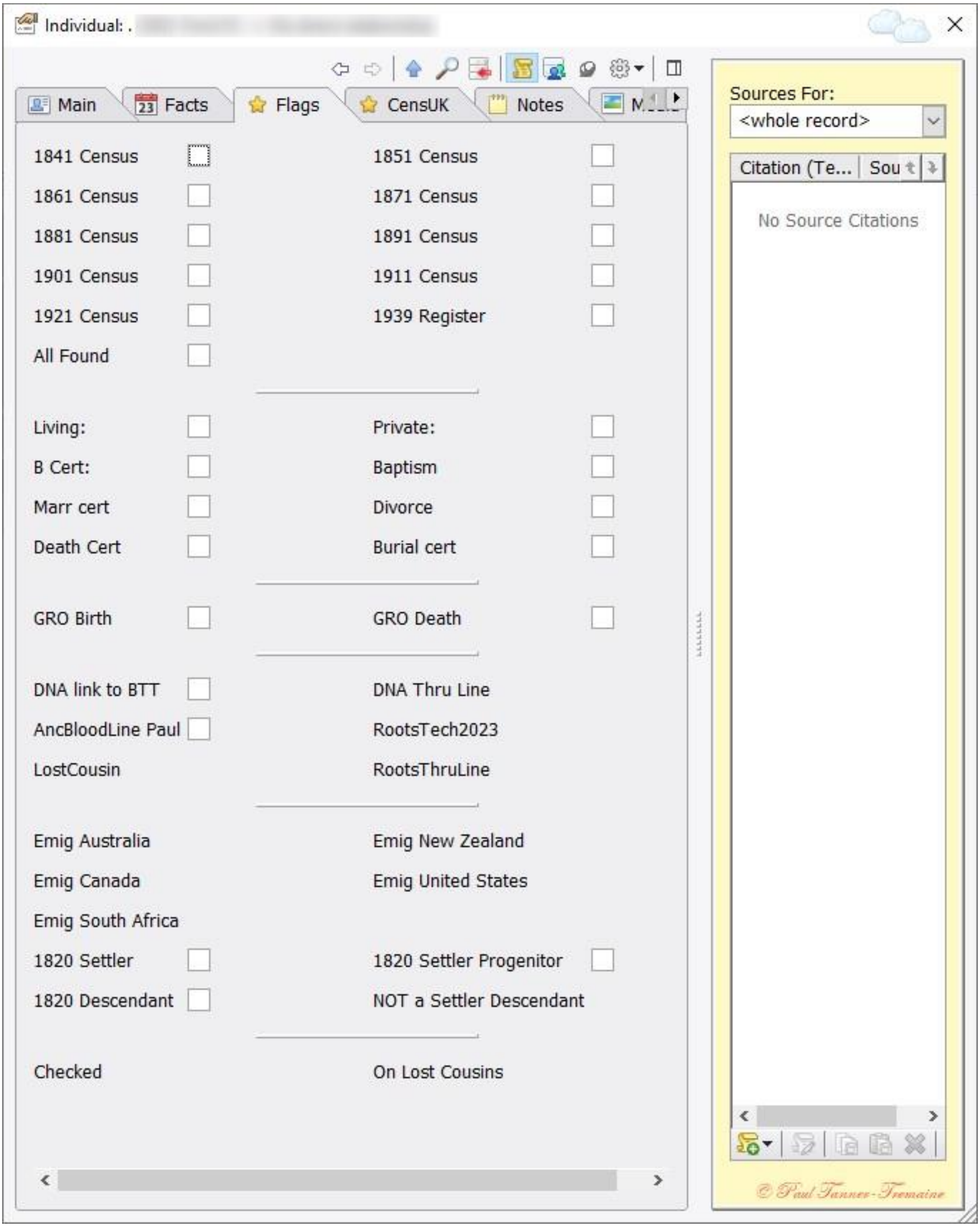
Sex: Male

Sources For: Birth

No Source Citations

Add Child

My Property Box Flags Tab layout



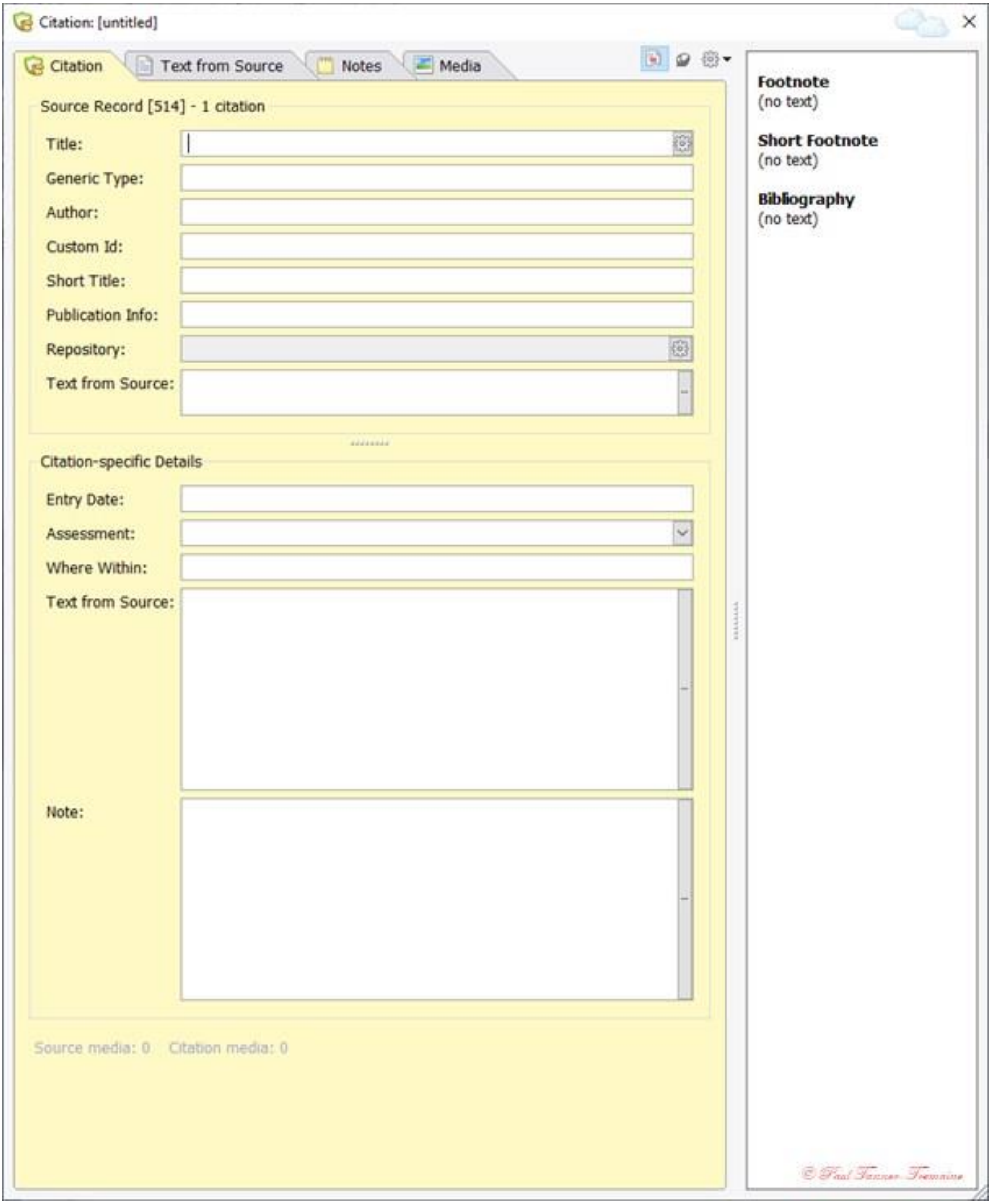
1841 Census	□	1851 Census	□
1861 Census	□	1871 Census	□
1881 Census	□	1891 Census	□
1901 Census	□	1911 Census	□
1921 Census	□	1939 Register	□
All Found	□		
Living:	□	Private:	□
B Cert:	□	Baptism	□
Marr cert	□	Divorce	□
Death Cert	□	Burial cert	□
GRO Birth	□	GRO Death	□
DNA link to BTT	□	DNA Thru Line	
AncBloodLine Paul	□	RootsTech2023	
LostCousin		RootsThruLine	
Emig Australia		Emig New Zealand	
Emig Canada		Emig United States	
Emig South Africa			
1820 Settler	□	1820 Settler Progenitor	□
1820 Descendant	□	NOT a Settler Descendant	
Checked		On Lost Cousins	

Sources For:

<whole record>

No Source Citations

My Generic Source and Citation layout



Citation: [untitled]

Source Record [514] - 1 citation

Title:

Generic Type:

Author:

Custom Id:

Short Title:

Publication Info:

Repository:

Text from Source:

Citation-specific Details

Entry Date:

Assessment:

Where Within:

Text from Source:

Note:

Source media: 0 Citation media: 0

Footnote
(no text)

Short Footnote
(no text)

Bibliography
(no text)

© Paul Tanner-Tremaine