What you may need to modify for your FH

23 September 2023 16:01

This document is based on my Generic BMD Auto Type template v1.10

Text Templates
You should only need to create new text templates if you create a *var.fieldname* and want to hide it from the control panel. I currently have 56 Text Templates, and all bar the first 4 will be hidden from the ORA Control Panel. All the var.* text templates are fields that I use internally in the Auto Type Template and associated libraries, or in the case of numbers 5 through 12, as result fields for the user responses to prompts. Click the Legend link to see all.

Auto Type Templates

Button 1
Changes you should/could make to this template are minor

a. The line [=:Assessment_U_Q_S_P:P] defaults to the parameter P for Primary assessment. If you rate the source of this collection differently, say it has no images to verify the accuracy of the transcription, you could change the parameter to S for Secondary. If you don't use the assessmnets in your sourcing, you could comment out the line with a # in the first column.
b. Immediately after the lib.Initvars is another library [lib.CheckEventType] which does a display of the 'calculated' zEventType and asks for a verification that it is correct. If you Cancel here, the autotype aborts, the OK will let it run to completion. Added this to stop 'run-away' autotypes where the incorrect EventType was being processed.
c. The template prompts for Set Event Flags. A response of 'y' or 'Y' will cause the SetFlags library template to run and update my Properties box Flags tab; any other entry will default to not running the SetFlags. If you generally don't set flags, you could remove the prompt line, meaning it will always default to a no condition and still run the [lib.SetFocusonEvent:var.zEventType]

Button 2
Use of this button will create a residence fact for an individual where the focus is on their Property Box Name field. Should not need any changes. It is useful in the situation where you have a Child baptism and you want to record the Parents' Residences at the time of that baptism. After you have run button 1

for the Child, move up to a parent and then run this button2. It will prompt for the residence, so fill it in as complete as you like.

Button 3
Use of this button, very similar to button 2, will create an Occupation fact, including their residence for an individual. Typically, after having processed a baptism for a child with button 1, move to their father's record and run this button 3. Should not need any changes.

Button 4
I wrote this as a result of finding a marriage record for an individual I had in FH, I found that I did not have the Spouse. You may need to adjust the number of {tab}s to get to the add spouse prompts.

Button 5
This is another auto type button I wrote out of necessity, and I should make this Button top priority! It checks to see if I have the Source.Title or Collection in my Generic source's list. If it comes up with the source in the selection box, just Cancel, you're good to run button 1. If it comes up blank, Cancel, and then run button 6, which will create the basics for the Generic Source, and then manually close that Source pane, go back to your Indi's Property Box name, and then run button 1.

Button 6
If you read button 5's description, then you know what this does. Nothing to change here.

Button 7
Again, written out of necessity where I had forgotten to add an image to a Citation, or had an old source/citation where I could now add the image as reference. See the comments under the library template
[lib.WriteImageToCitation:var.zEventType]

Button 8
As I have not yet found a successful method of moving the cursor on the Generic Source/Citation Media tab to invoke the edit button for the Image, I do that manually and then this button 8 will fill in the relevant Image date, URL and keyword to the media record.

Library Templates

[lib.AddNewGenericSource:var.zEventType]

You may need to change some of this if your generic source layout has changed, or if it is not the same as mine.

[lib.CheckEventType:var.zEventType]
If you add a new 'collection type' then this library element will need to be updated as well. It show the value of zEventType that has been determined in lib.Initvars and prompts for an OK or Cancel button click. It ensures that the rest of the autotype will run with the correct Event type for the Service Collection record that you are processing.

[lib.PathToImageFilename:var.zEventType]
Should only be needed to set up once, to define your folder structure and naming convention for filenames. It is called by text Template #3 Image filename
This library sets up the path to where the image can be saved. The number of \'s and leading or trailing spaces are crucial to get the right path!

var.p1 sets up where your FH Media folder resides on your PC.
var.p2 is the subfolder under ..Media
var.p3 I use the naming convention of Surname, Given names, and if a Marriage event, add the spouse's name as well
var.p4 the year of the event
var.p5 the keyword category I assign to the last part of the file name
var.p6 is the file type, ie .jpg

See also [lib.WriteImageToCitation] below.

[lib.Initvars]
This library is one of the key libraries behind my Generic BMD Auto Types. Its initial creation came about in an attempt to overcome the variants in field names used by different collections within a Service and across the Services. For example FamilySearch.org has not yet made up their mind whether to call a date of baptism, a Baptism Date, Baptism date, or Event Date, etc. Ancestry seems to use Baptism Date, as does Findmypast, but I have seen variants. I have tried to consolidate all these variants into one var.zEventdate. Same with Places, to apply the variants as var.zEventplace.
To cover all collection types, I have a var.zEventType, etc, and this made it easier to code the templates and libraries, and it is used as a parameter to other library templates.
Initvars has been structured

a. firstly by the Service in [o.Service] types, and within each [o.Service] an initial section of [var.z*Fielddefinitions*] which are common to all the collection or EventTypes within that service.
b. Secondly by the determination of an Event Type, according to fields that ORA has found within that Collection, creating the value for

var.zEventType.

c. Thirdly, by the value of var.zEventType, which includes the specific anomalies found with variants within that var.zEventType.

This structure is repeated for each of the Services, currently Ancestry, FamilySearch, FindmyPast, Freereg and Find a Grave. To add a new Service, you will need to create a Service section, with all the collection or event types within that section. If you can use the same var.*zFielddefinitions* that currently exist, then most of the auto types will run without change.

[lib.SetFlag:var.zEventType]

I have a Flags tab set up on my Properties Page that shows all the flags that I may have set for an individual. These flags can be used in Diagrams/Charts and queries to show or filter results.
This library moves the focus to the Flags tab, and then tabs down the configured number of {tab}s and uses the {space} to toggle on or set the flag, based on the var.zEventType value. Once it has done that, the focus moves back to the Main Tab, and calls the [lib.SetFocusonEvent] to move the focus to the relevant event fields.
You may have to change the tabs count to match if you use a Flags page.
If you don't have or use Flags, see the note above in Auto Type Button 1 about flags.

[lib.SetFocusonEvent:var.zEventType]

This simple library template just moves the focus point on the Properties page Main tab to the relevant field from the var.zEventType. Change the {tab*nn } counts to match your Main tab field layout.

[lib.WriteFactData:var.zEventType]

This library template assumes that the [lib.SetFocusOnEvent] has run and positioned the cursor in the correct date entry field. Depending on the value of the parameter var.zEventType it will post the relevant date and place in the fields on the Main Tab, or create a fact entry if needed. See the entry line for Christening. (new to come will be Probate)

[lib.WriteSource:var.zSource]

This library template create a new generic source. I currently {tab*8} over the fields, but you could break these tabs up and post relevant info in the Generic Source fields.

[lib.WriteCitation:var.zEventType]

This library is very similar to Initvars in its structure, by Service and EventType. It is the place where all the Citation Text Form Source data to be output is created, effectively into a consistent template for each time the Generic AutoType is run.
This is where most of the changes will begin, as you add or update the way you

would like to see the Text From Source displayed in your generic Source and Citation. If you find a new Service other than the ones already configured, you will need to add a section for that Service and its Collection or Event Type, following the similar structures already defined. There is a standalone section at the bottom, which changes the TFS entry field from standard text to the Rich Text format and adds the hyperlinks to the record and the image from the source

[lib.WriteImageToCitation:var.zEventType]
Should only be needed to set up once, to define your folder structure and naming convention for filenames.
This library sets up the path to the saved image, and then attaches the image to the generic citation Media Tab. The number of \'s and leading or trailing spaces are crucial to get the right path!

p1 sets up where your FH Media folder resides on your PC.
p2 is the subfolder under ..Media
p3 I use the naming convention of Surname, Given names, and if a Marriage event, add the spouse's name as well
p4 the year of the event
p5 the keyword category I assign to the last part of the file name
p6 is the file type, ie .jpg

For Example the result should be similar to:
E:\\My Drive\\Family Historian Projects\\MyFamily\\Myfamily.fhdata \\baptisms\\Bloggs, Joe 1789 bapt.jpg
Once I have built the path to the image, I manually move to the Citation Media tab before running the autotype.

[lib.SetKeywords:var.zEventType]
For each event type, I have a keyword which I add to the image Keyword field, to assist in searching or filtering the images. Update or add a new row if you have different keywords

Other Library Templates that are needed but may not need any changes

[lib.Dividerbar]

[lib.SetAssessment]

[lib.AddRecordLinktoTFS:var.zRecURL:var.zURLhrefdesc]

[lib.AddImageLinktoTFS:var.zImageURL:var.zImagehrefdesc]

[lib.AgeCleanup]

[lib.Add_FH_Fact:*Residence*:var.zEventdate::var.Their Residence]

[lib.Add_FH_Fact_with_desc:*Occupation*:var.Their Occupation:var.zEventdate:var.Their Residence]