

Blueprints

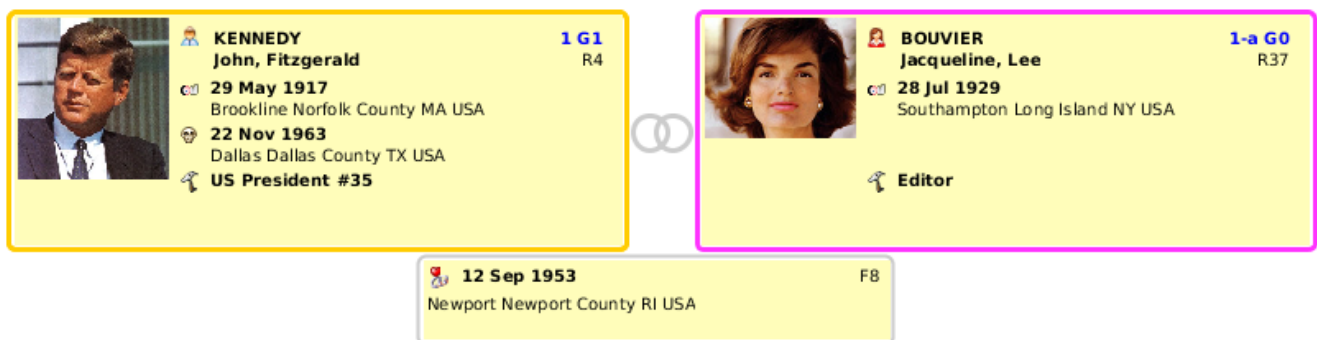
A blueprint is description used to customize the display of an entity in certain views of Ancestris.

Views using blueprints

Only the following views use blueprints. For each of them, an example shows what the blueprints could look like.

Dynamic tree

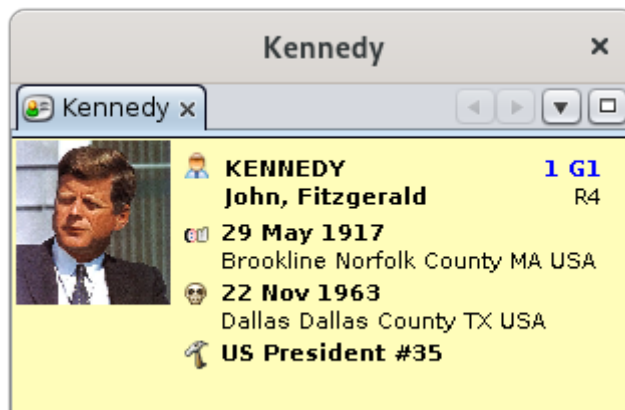
The **Dynamic Tree** uses blueprints for individuals and families. Blueprints are the areas displayed inside the coloured frames.



Entity card

The **Entity Card** uses blueprints for all categories of entities.

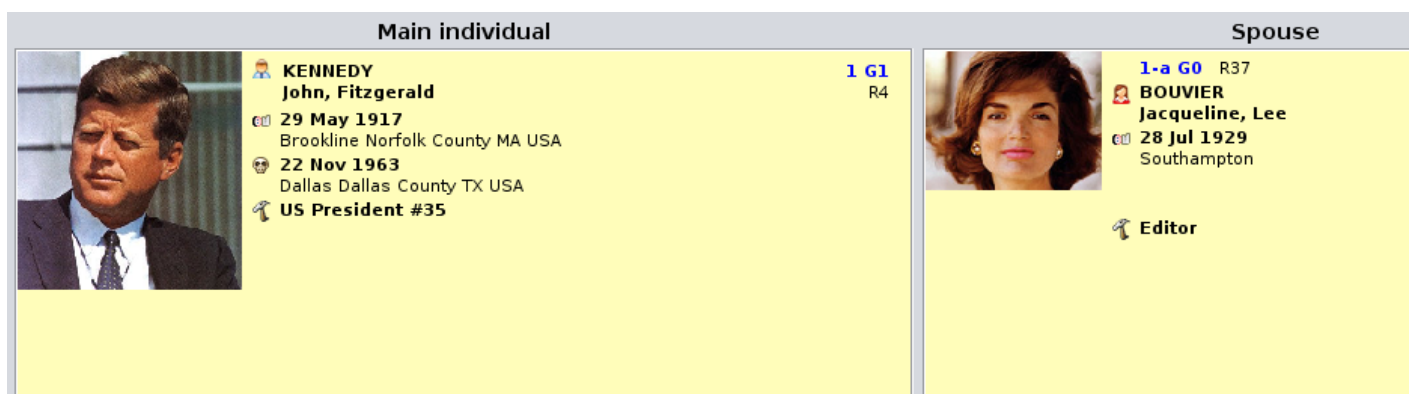
Blueprints can be reused for several views. Here, the blueprint used for individuals in the dynamic tree is used inside the Entity card.



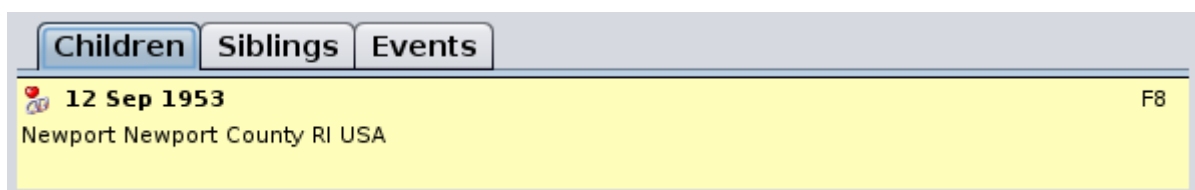
Extended navigator

The **Extended Navigator** uses blueprints for both individuals and families.

Below are blueprints used for the individuals of the selected couple. You may notice that they are not the same. The spouse blueprint does not display the Sosa number and displays the entity number above the name.



And this is a blueprint for each child of the selected family.



Gedcom editor

The **Gedcom Editor** uses blueprints in its lower part for all entity categories.

As for the entity card, the Gedcom Editor uses blueprints for individuals or families, but also for other entities.

For example here is a simple blueprint of a Repository entity.

Or a simple blueprint of a Multimedia entity.

Multimedia id=M00025



Now that you have seen what blueprints look like and where they are used, let us see how they work.

General principles

A blueprint is a template defining what to display of an entity, and how to display it.

A blueprint is assigned to a given entity category (individual, family, source, etc.) for a given view (Dynamic tree, Entity file, etc.).

You can define your own blueprints rather than using the ones already provided within Ancestris.

Several blueprints exist within Ancestris and are already selected by default for each of the entity categories, and in each view where blueprints apply.

Blueprints constitute a way to customize Ancestris. You are free to keep blueprints as they are or modify them to your needs.

You can define your own blueprints, i.e. both the information to display for an individual or a family (name, address, etc.) and how to display this information (colour, font, alignment, layout, etc.).

If you want to make some changes to the available blueprints, you will need to learn a little more about them. This is the subject of the following sections in this page.

Technically, a blueprint is a piece of HTML code written into a file located on your disk. Refer to the paragraph **Directory containing blueprints files** for more information.

Managing blueprints

Managing blueprints consists in performing two types of activities.

- **assign blueprints** to entities and views
- **build your own blueprints** or delete them

This is done using the **Blueprint window**.

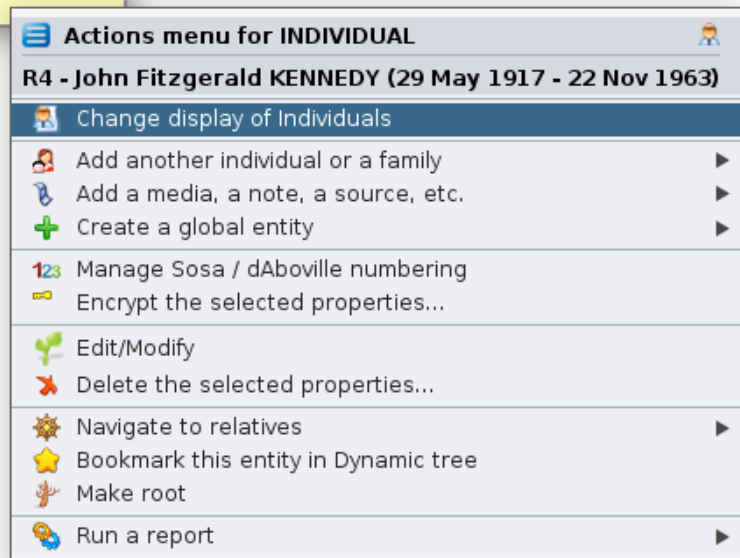
The Blueprint window manages the blueprints for one category of entity at a time.

The window is the same regardless of the category of entity, but it only contains blueprints attached to the given category of entity.

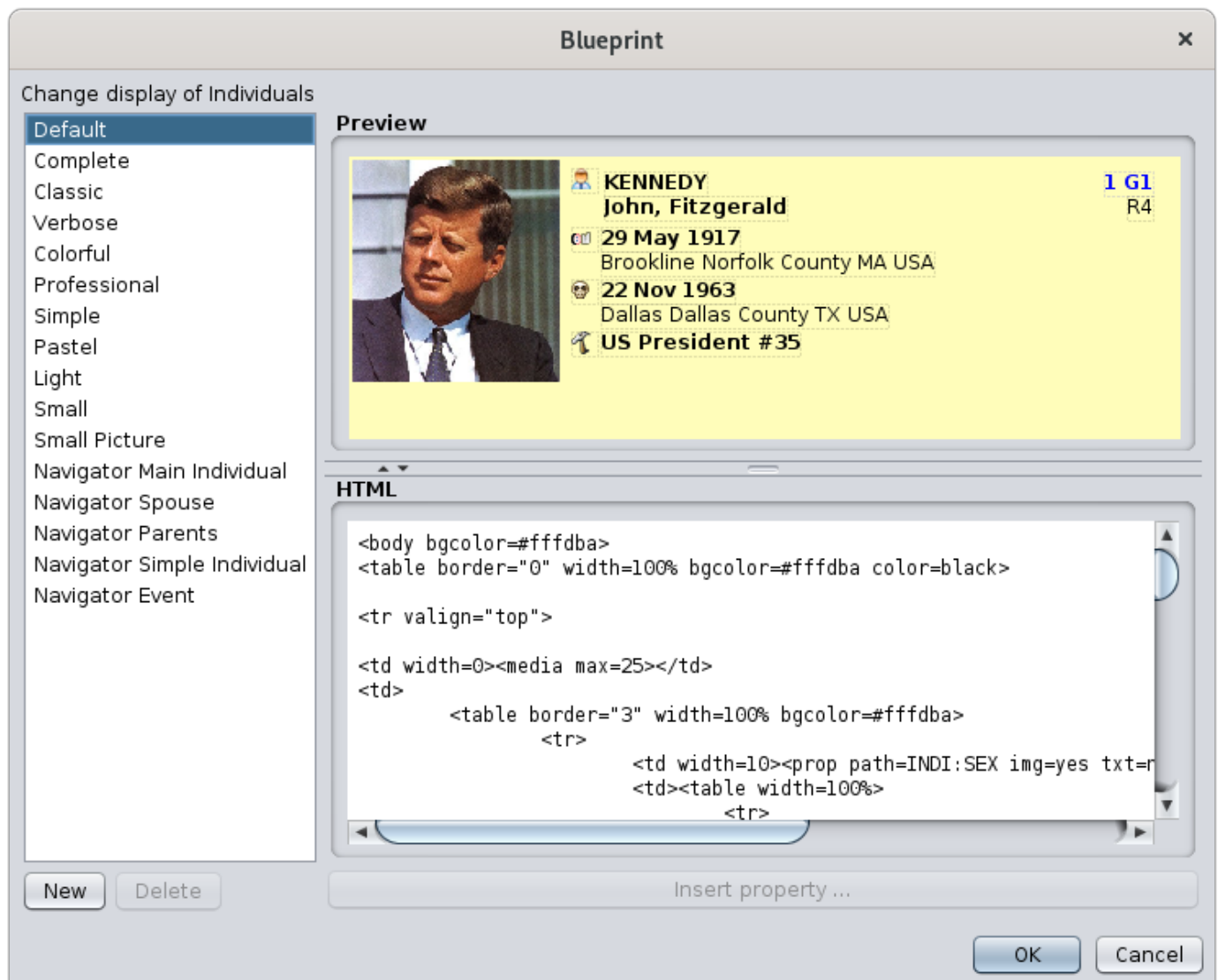
To access the Blueprint window for a given entity category, you must use the Ancestris context menu from an entity of the corresponding entity.

- Right-click on an individual in the Entity card view, then select "Change display of individuals" to open the Blueprint window for individual.
- Right-click on a family in the Entity card view, then select "Change display of families" to open the Blueprint window for families.
- Ditto for Notes, Sources, Multimedia, Repositories, etc. This can be done from the Entity card or the Gedcom Editor for the corresponding entities.

Here is how to access the Blueprint window for individuals, from the Entity card, Contextual menu by right-clicking, then choose "Change display of individuals".



The Blueprint window for Individuals appears.



Whatever the category of entity, the operations in this window are the same. We will use the example of the Blueprint window for individuals in the description that follows.

Description

The Blueprint window contains the following areas.

- The list of blueprints on the left shows the blueprints currently available for the previously selected entity category. To work on a blueprint, select it from the list.
- The preview area at the top right shows a preview of the blueprint selected in the list.
- The HTML editor at the bottom right displays the HTML code of the blueprint selected in the list on the left, and where you can edit the code. The horizontal separator bar under the preview area must be dragged up with the mouse to see the code below.
- Meaning of the buttons:

- The **New** button adds a copy of the currently selected blueprint to the list, and asks you to input another name. This will become the newly selected blueprint and you will be able to modify it.
- The **Delete** button removes the selected blueprint from the list
- The **Insert Property...** button appears when you edit HTML code. Use it to insert a TAG in the HTML code. It will insert it at the cursor position.
- The **OK** button validates the changes and exits the window.
- The **Cancel** button closes the window without validating the changes.

The Blueprint window size can be changed by pulling the lower right corner.

The preview area and the Text Editor area can be adjusted by moving the separator of these two areas with the mouse or by clicking on the arrows of the separator on the left:



Usage

The Blueprint window covers the following needs:

- **Discover the available blueprints.** To discover the available blueprints without activating them (therefore, without closing the panel), click successively on each of them in the left-hand side list, and check the result in the preview area.
- **Choose a different blueprint.** To assign another blueprint to the currently selected entity category in one of the views, click a blueprint in the list, then validate by pressing the OK button.
- **Modify a blueprint.** For security reasons, the blueprints already available cannot be modified. To modify an existing blueprint, you must first make a copy of it. Select it in the list, click New and give it a new name. Then use the HTML editor panel to make your changes.
- **Create a blueprint.** To create a new blueprint, it is recommended to copy an existing one. See above.
- **Delete a blueprint.** To delete a blueprint, select it in the list and click the Delete button. You are then asked to confirm the deletion which will be final. Note: for security reasons, blueprints provided with Ancestris cannot be deleted.

Case study

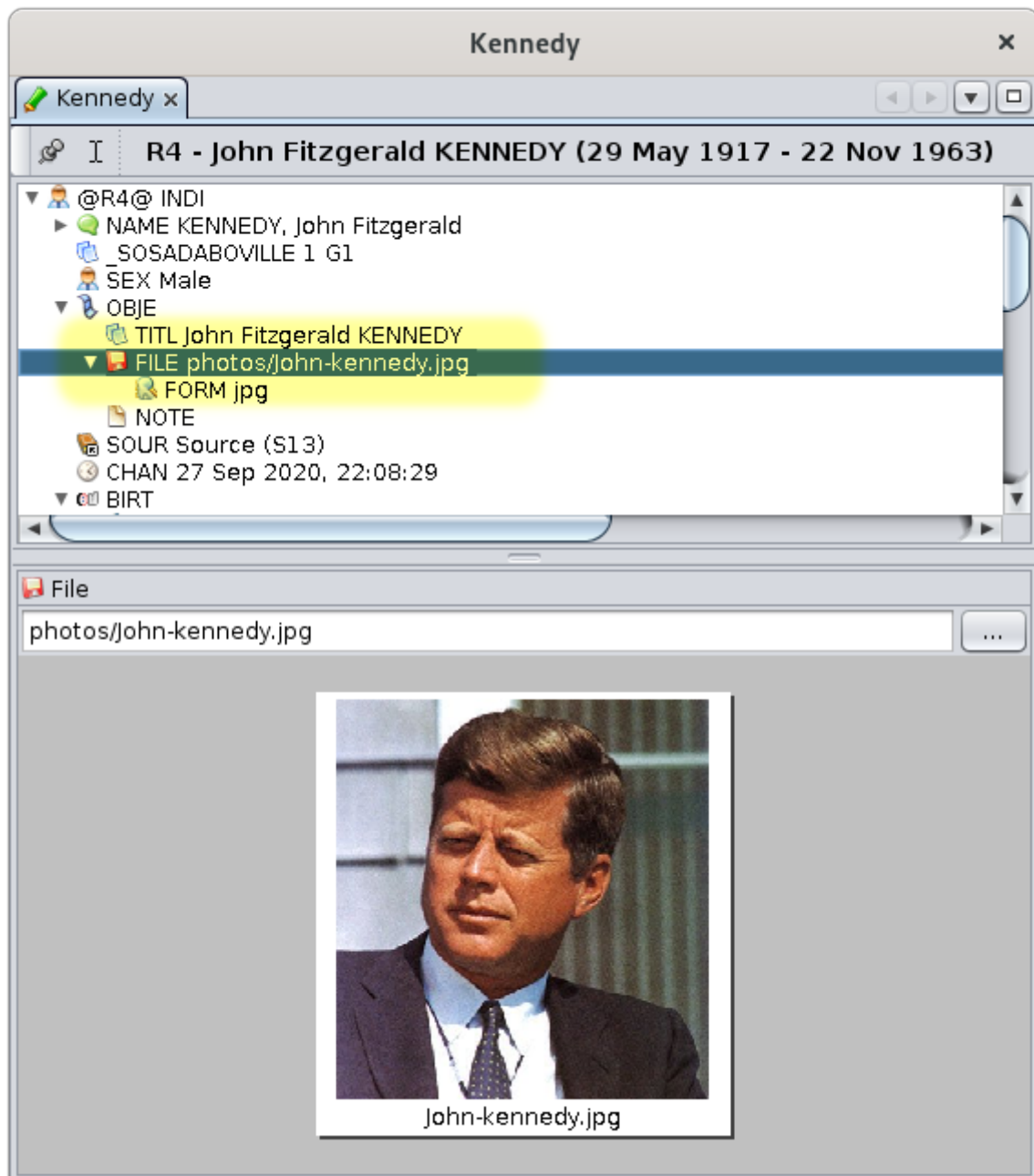
Suppose we have a simple blueprint without a photo for the individuals in the dynamic tree, and we want to insert a photo. The steps to do this would be the following:

1. **Identify where the photo data item is** inside the data of the individual. Use the Gedcom editor for that.
2. **Open the Blueprint window for individuals**
3. **Create a new blueprint** by copying 'simple blueprint' without image. Rename it 'MyBlueprint-Indi-Tree'
4. **Modify the blueprint** to insert the photo property identified above, and adjust its size in the frame using the preview
5. **Save the blueprint and check the result**

Each of these steps is detailed below.

Identify where the photo data item is

The **Gedcom editor** will show you where the photo items are located for all individuals. We see below that the photo is located in the FILE tag, attached to the OBJE tag, which itself is attached to the INDI tag. All together, you will need to remember that photo files are at the address '**INDI:OBJE:FILE**'. That is what we wanted to know.



Open the Blueprint window for individuals

As we want to change the blueprint for Individuals in the Dynamic view, we need to first select an individual in the Dynamic view.

Then we right-click on the individual. In the **context menu** that appears, we choose "**Change display of individuals**".

The screenshot displays a software interface with several individual profiles and an open actions menu.

Individual Profiles:

- KENNEDY John, Fitzgerald** (R4): 29 May 1917, Brookline Norfolk County MA USA; 22 Nov 1963, Dallas Dallas County TX USA; US President #35.
- BOUVIER Jacqueline, Lee**: 28 Jul 1929, Southampton Long Island NY USA; Editor.
- SCHLOSSBERG Edwin** (F68): abt 1945.
- KENNEDY John, Fitzgerald** (partial): Nov 1960, Washington DC USA; Jul 1999, Martha's Vineyard MA USA; New York City.

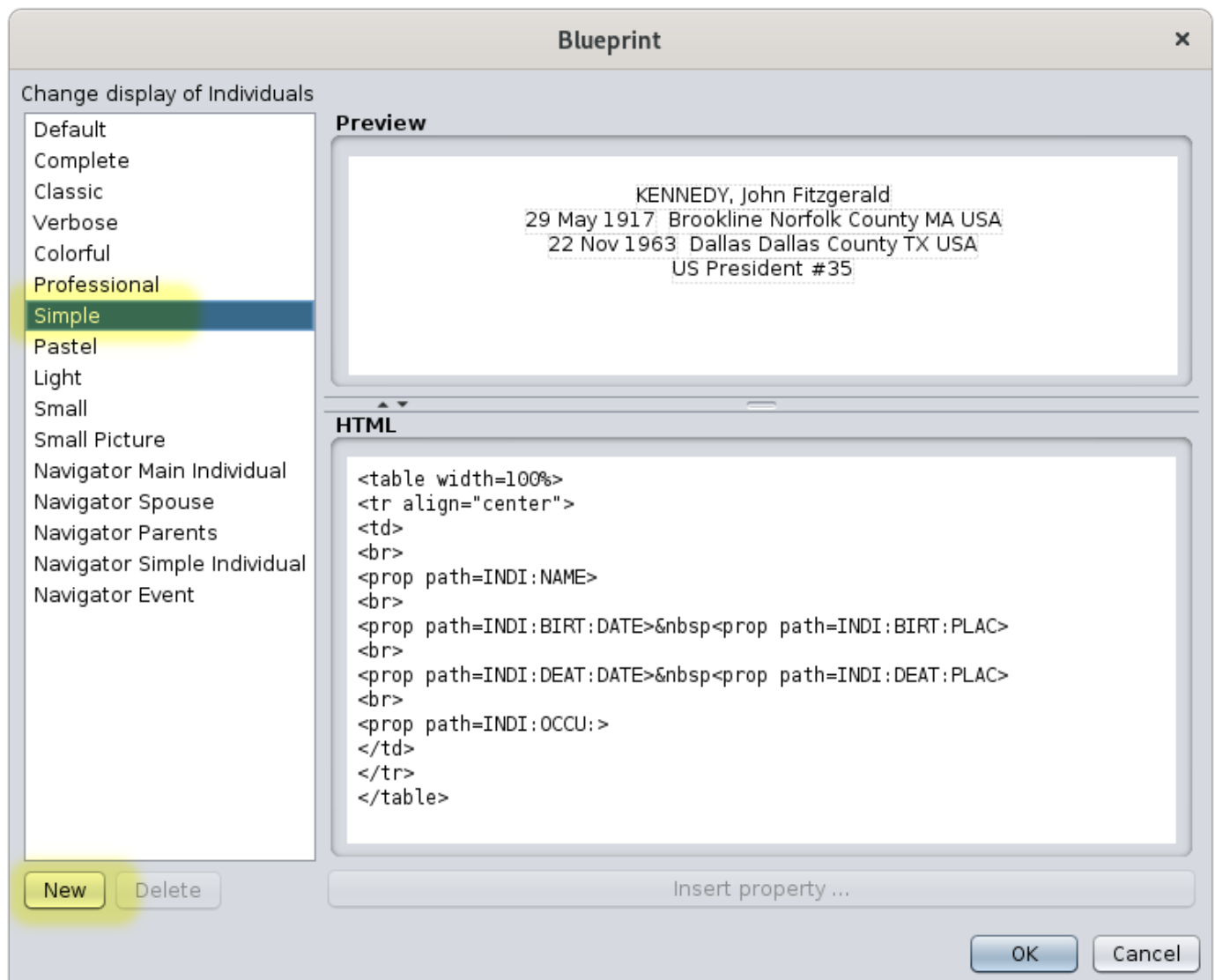
Actions menu for INDIVIDUAL:

- R4 - John Fitzgerald KENNEDY (29 May 1917 - 22 Nov 1963)
- Change display of Individuals
- Add another individual or a family
- Add a media, a note, a source, etc.
- Create a global entity
- Manage Sosa / dAboville numbering
- Encrypt the selected properties...
- Edit/Modify
- Delete the selected properties...
- Navigate to relatives
- Bookmark this entity in Dynamic tree
- Make root
- Run a report

A callout box labeled "Change display of Individuals" points to the corresponding menu item.

Create a new blueprint

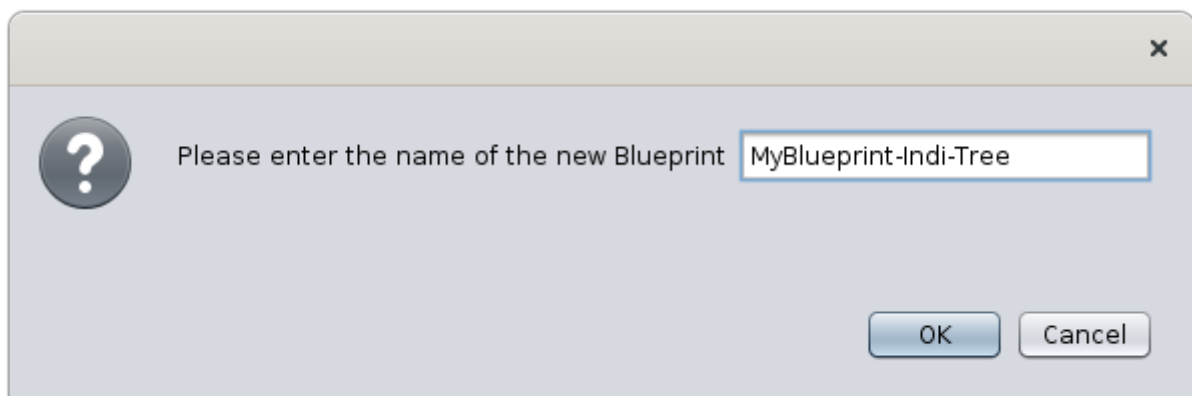
The Blueprint window for Individuals appears.



As we want to make a new blueprint as an extension of the "Simple" blueprint, we first make a copy of the "Simple" Blueprint.

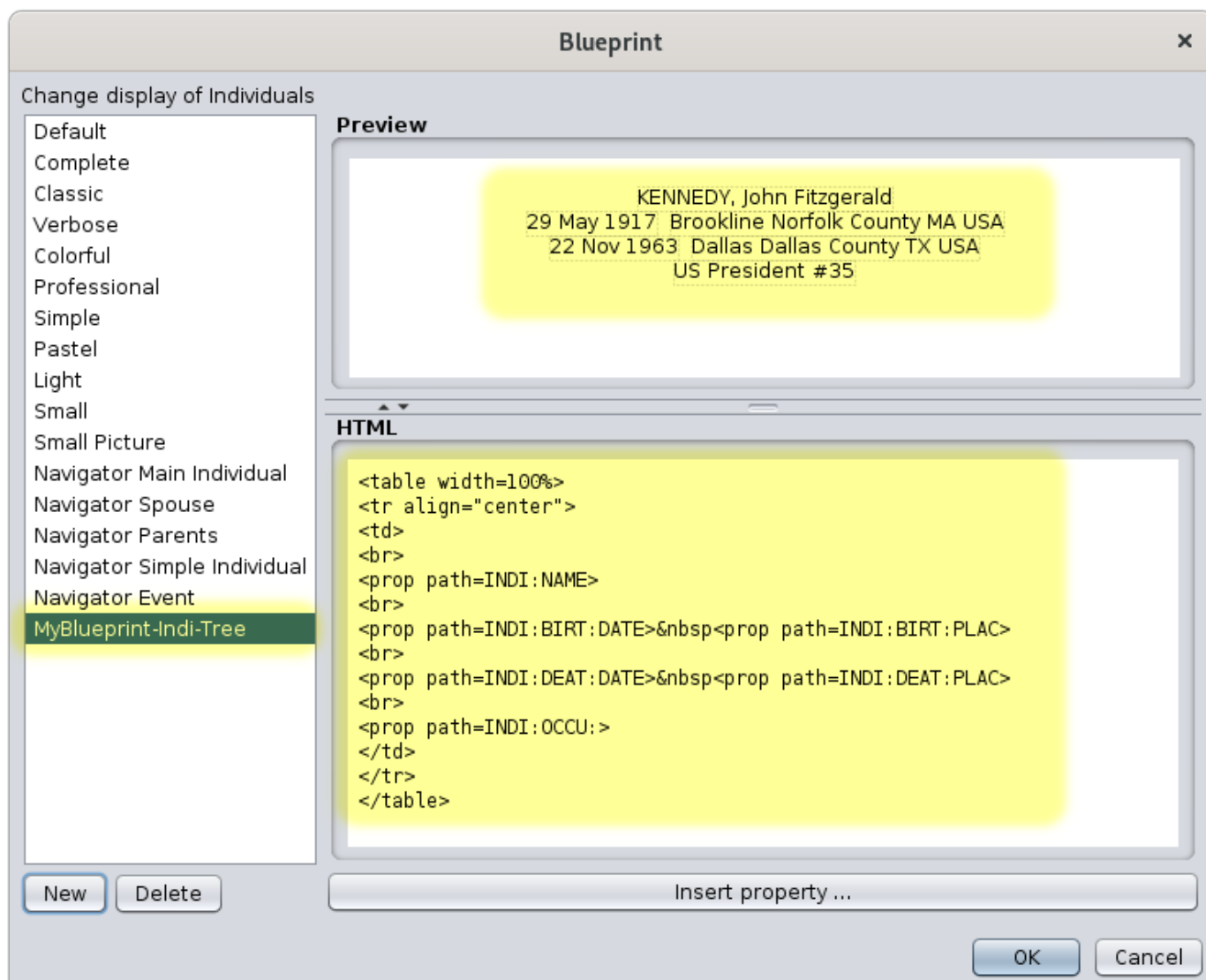
To do so, we select it in the list. Then in the preview, we see what this Blueprint displays for an individual : simple text without a photo. What we want to do is to add the photo of the individual.

We then click "**New**". Ancestris asks the name of the new blueprint:



We type 'MyBlueprint-Indi-Tree' and press OK.

We are then on the previous Blueprint window with a new editable blueprint.



We have the new blueprint selected in the list on the left, the same preview at the top right, and the HTML code at the bottom right.

We see that this blueprint is editable, because the buttons "**Delete**" and "**Insert property...**" are visible and enabled.

We will now change the HTML code so that it displays the photo.

Modify the blueprint

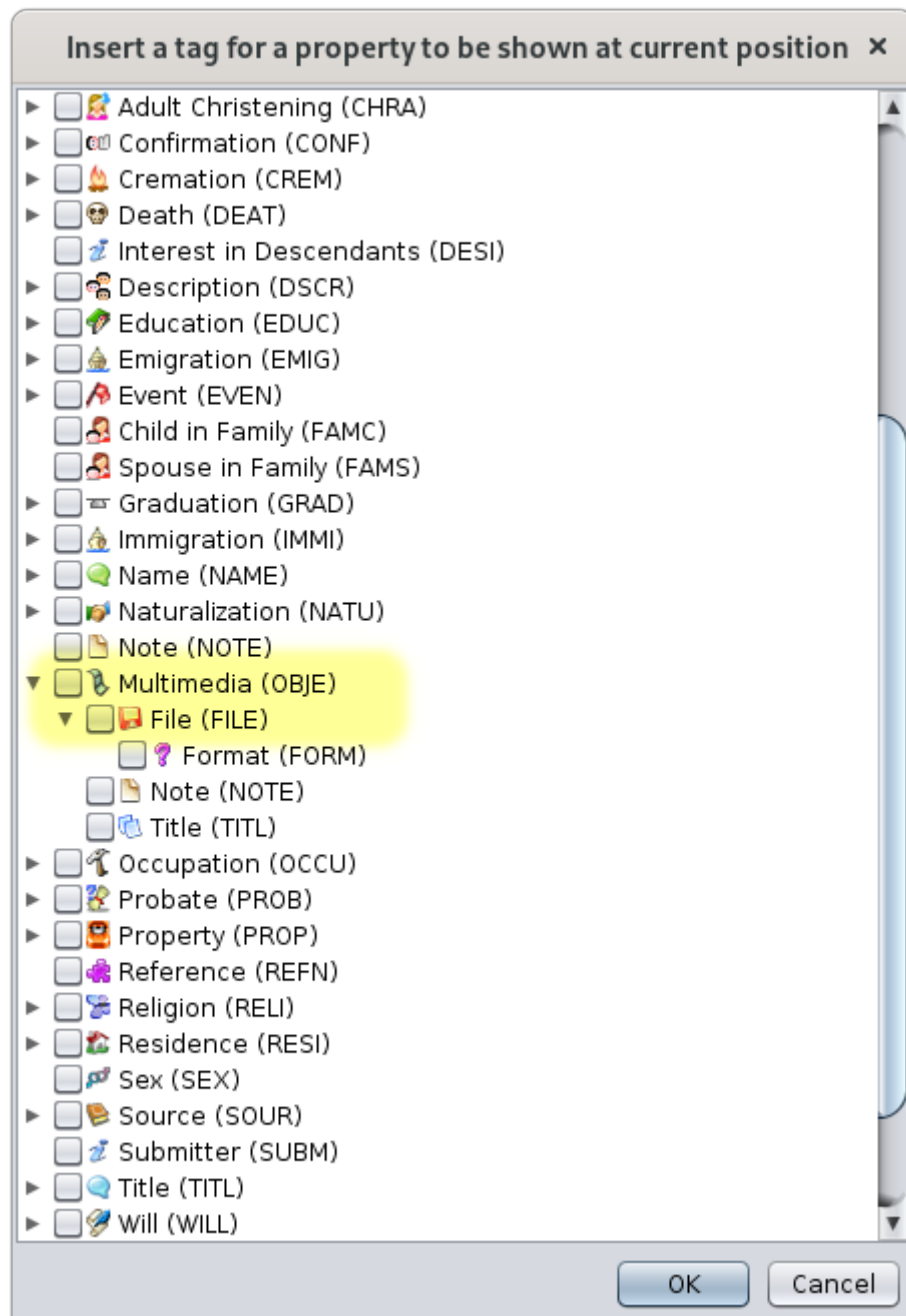
We can see that there are 4 lines of properties in the HTML code: name, birth, death, occupation.

The HTML <prop> tag, and everything in it, is Ancestris specific. This makes it possible to manipulate the genealogical data of the Gedcom file.

We want to put the photo at the bottom of the display, so we insert another property, the one identified above, INDI:OBJE:FILE, after the 4 properties already showing.

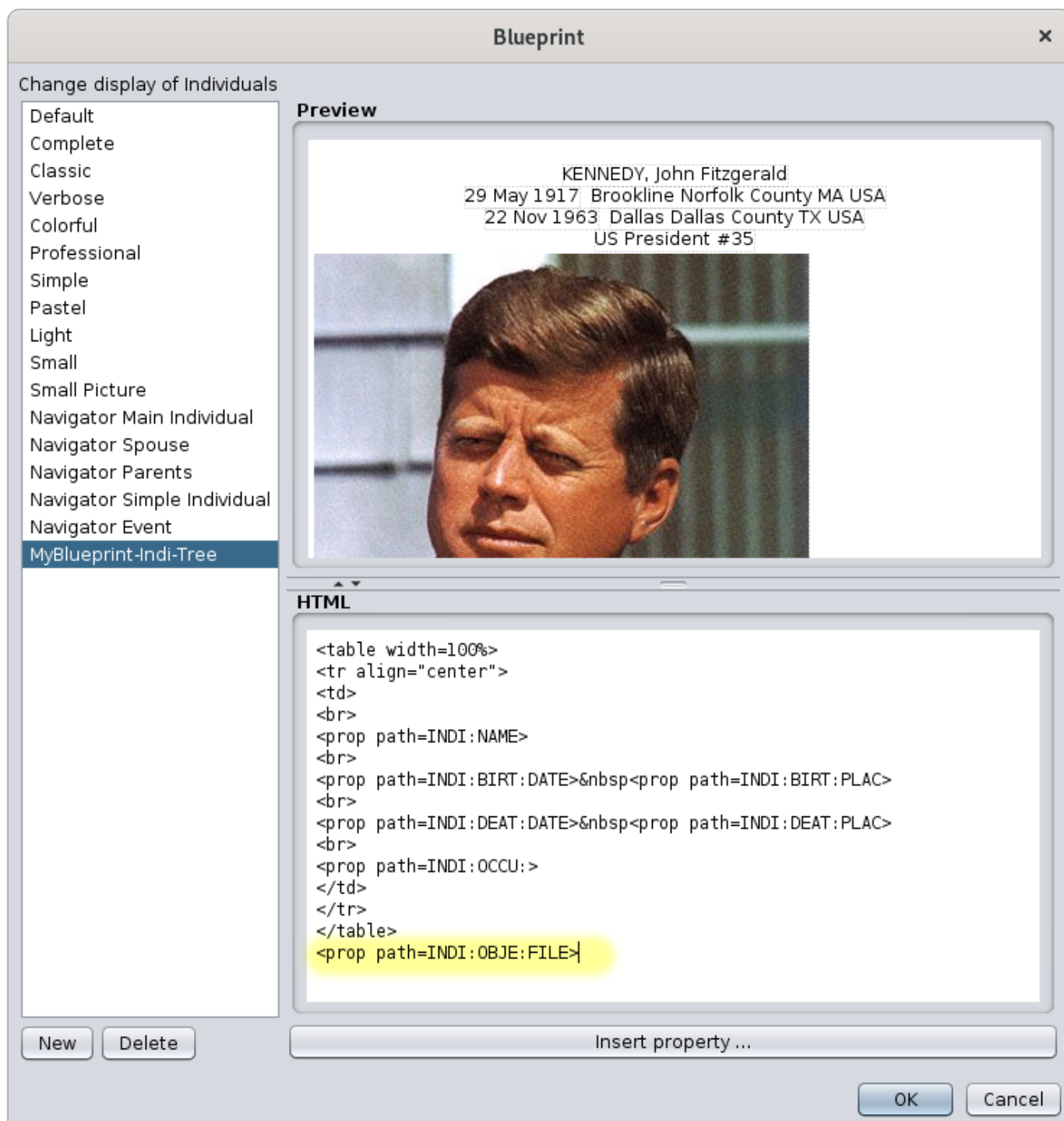
We place the cursor under </table>, the label that marks the end of a table.

We press the button "**Insert property...**". A window appears listing a choice of possible properties.



In this window, we find the Multimedia OBJE line, we unfold the flap to show the content of OBJE, and we see the line File. We check the box. And we press "OK".

We now see that the preview has changed.

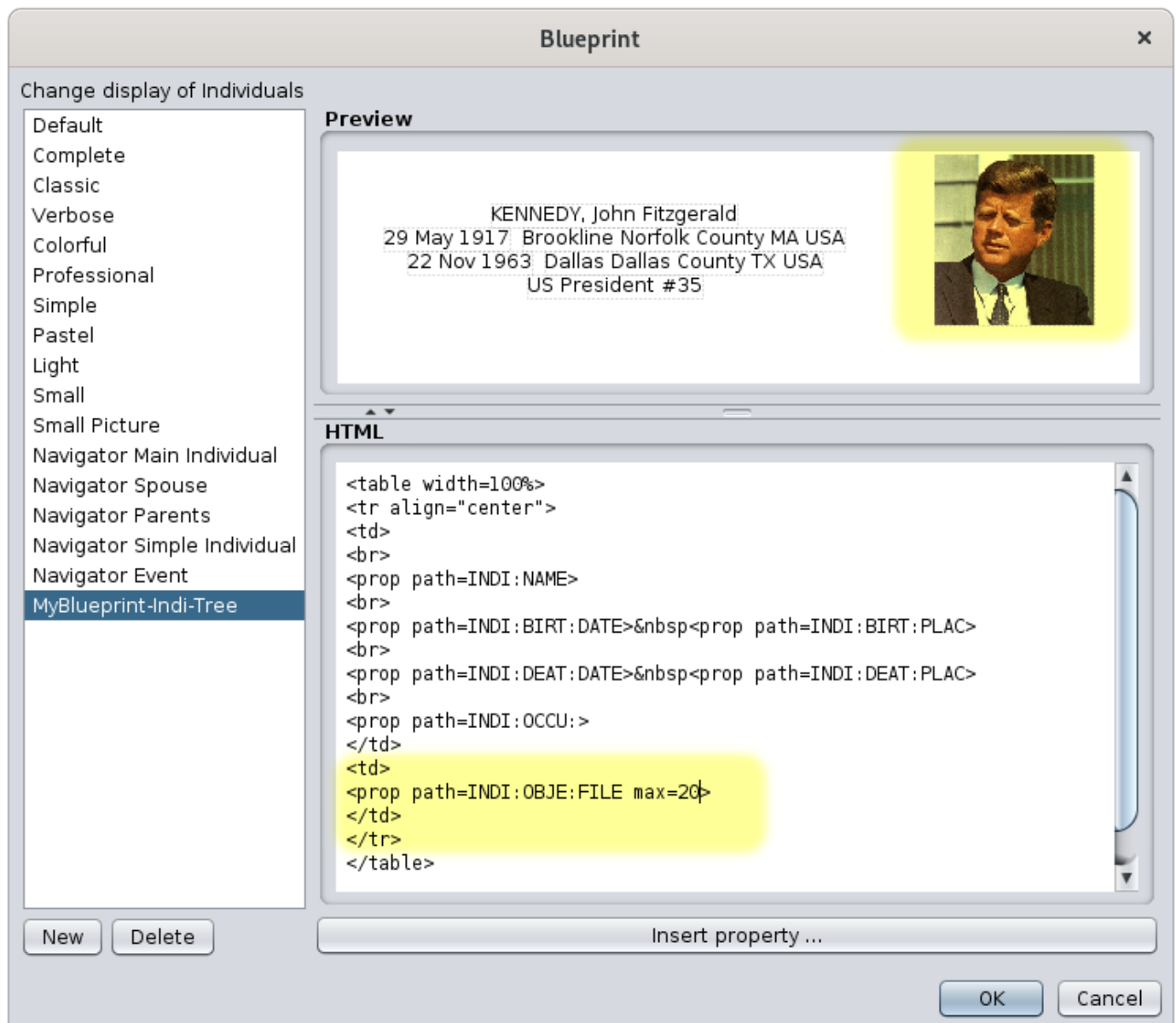


But that's not finished yet, because in the preview you can see that the photo is too big. And we changed our mind as well. We want to position the photo at the top right corner of the text.

As the text was structured inside a table, it is in fact necessary to put the image in the table, to the right of the text. To do this, we modify the HTML code as follows: we move the line that we have just inserted in the table structure by adding a cell using the `<td>` `</td>` tags.

As we also want to limit the size of the image, we add an Ancestris specificity: we add "max = 20" at the end, inside the `<prop ...>` tag, which means to limit the photo size at 20% of the available

width. The preview changes as you change the code. This is handy and immediately lets you see if what you are doing is right. Once the code has changed, the preview looks correct, and the window looks like this.



The code in yellow does its job: the photo is in a cell on the right of the table and limited to 20% of the width.

Save the blueprint and check the result

We can save our work now. We click on OK and we look at the dynamic tree at the same time. The blueprint changes.



It's good, we have what we wanted!

Going further

Of course, the hardest part therefore remains to know HTML on one hand, and to know the specificities added by Ancestris to handle genealogical data on the other hand.

As far as HTML language is concerned, there are many sites to help you. Please refer to them.

Regarding Ancestris specificities, the tags added to the HTML are:

- `<prop>` or `<ifvalue>`
- `<name>` or `<i18n>`
- `<media>`
- `<mark>`

The following sections describe how to manipulate them.

`<prop>` or `<ifvalue>` tags

```
<prop | ifvalue path="path:to:prop[?i][:end:of:prop]" [default="path:to:other:prop"]
[format="format"] [max="n"] [length="n"] [img="yes"] [txt="no"] [underline="yes"]>
```

Displays the property of the entity defined by the **path**. If a **default** path is also specified, it is the one that is used if the first property is not found.

For **multiple properties**, a rank can be indicated in the path using "*i*" where *i* is the rank of the property. Following this number, the rest of the path can be provided if any. See example below with the occupations.

The **format**, **length** and **max** parameters are explained below.

The **img** parameter can be set to 'yes' or 'no' to display the property's icon. By default, it is 'no'.

The **txt** parameter can be set to 'yes' or 'no' to display the property name or not. Default is 'yes'.

The **underline** parameter can be set to 'yes' or 'no' to underline the property's text. By default, it is 'no'.

Parameters in square brackets are optional.

Either 'prop' or 'ifvalue' can be used.

Case of names (NAME property)

To customize the display of the name, you can use the name sub-properties.

Example : `<prop path=INDI: NAME > a. k. a <prop path=INDI: NAME: NICK>` will display the full name and the nick name.

Example : **John F. Kennedy a.k.a Jack**

Case of places (PLAC property)

It is possible to force the level of jurisdictions that you want to see displayed for a place.

The syntax of the line to put in the HTML editor is of the following form

```
<prop path=INDI: BIRT: PLAC format="1 ( 4) ">
```

where 1 and 4 are the jurisdiction ranks in the Place criteria definition (see **File/Properties**).

Example: assuming that all our places follow the following format (as per the PLAC tag in the Gedcom file or in File/Properties):

```
Suburb, City, ZIP code, State, Country
```

Starting counting from the left at rank #0 : rank #1 is the City and rank #4 is the country.

In this case:

- `<prop path=INDI: BIRT: PLAC format="1 (4) ">` displays **City (Country)**
- `<prop path=INDI: BIRT: PLAC format="1, 0">` displays **City, Suburb**
- `<prop path=INDI: BIRT: PLAC format="1, 3, 2 - 4">` displays **City, State, ZIP Code - Country**

Case of a multiple property, such as occupations (OCCU property)

In the case of multiple properties, it is possible to force the rank of the property to be displayed.

The syntax of the line to put in the HTML editor is of the following form

```
<prop path=INDI: OCCU?2>
```

This will display the second occupation of a person.

You can access to the last value with the LAST keyword. The last value can be the first of one person and the third for another one.

```
<prop path=INDI: OCCU?LAST >
```

If you want to display the date of the second occupation of a person, the code would be the following:

```
<prop path=INDI: OCCU?2: DATE>
```

It is therefore possible to display the first 5 occupations of every persons like this. In the Default blueprint, you would change the code and insert the 5 blocks like this:

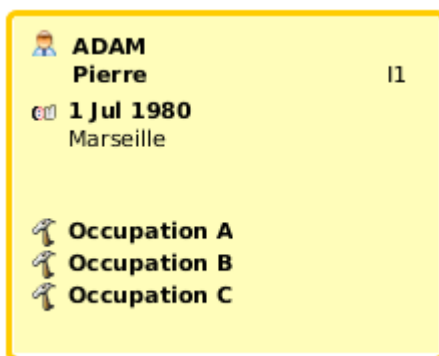
```
<tr>
  <td><prop path=INDI: OCCU?1 img=yes txt=no></td>
  <td valign="top">
    <b><prop path=INDI: OCCU?1 max=67></b>&nbsp;<br>
  </td>
</tr>
<tr>
  <td><prop path=INDI: OCCU?2 img=yes txt=no></td>
  <td valign="top">
    <b><prop path=INDI: OCCU?2 max=67></b>&nbsp;<br>
  </td>
</tr>
<tr>
  <td><prop path=INDI: OCCU?3 img=yes txt=no></td>
  <td valign="top">
    <b><prop path=INDI: OCCU?3 max=67></b>&nbsp;<br>
  </td>
</tr>
<tr>
```

```

<td><prop path=INDI: OCCU?4 img=yes txt=no></td>
<td valign="top">
<b><prop path=INDI: OCCU?4 max=67></b>&nbsp;<br>
</td>
</tr>
<tr>
<td><prop path=INDI: OCCU?5 img=yes txt=no></td>
<td valign="top">
<b><prop path=INDI: OCCU?5 max=67></b>&nbsp;<br>
</td>
</tr>

```

For a person with 3 occupations, and using the code above, you would see the following in the tree view for instance:



Text formatting possibilities [format="*format*"]

This format only applies to monoline text.

- `format="$D"` : displays the full date of an event
 - Example : `<prop path="INDI: BIRT" format="{ $D }">` displays **10 feb 1876** as the date of birth (BIRT)
- `format="$y"` : displays the year of the date of an event
 - Example : `<prop path="INDI: BIRT" format="{ $y }">` displays **1876**
- `format="$P"` : displays the full location of an event
 - Example : `<prop path="INDI: BIRT" format="{ $P }">` displays **Hyde Park,02136,Boston,Suffolk,Massachusetts,USA**
- `format="$p"` : displays the city of the event
 - Example : `<prop path="INDI: BIRT" format="{ $p }">` displays **Boston**

- `format="$n"` : displays the name of the related entity if it exists
 - Example : `<prop path="INDI: CHR: XREF" format="{ $n}">` displays **Marie STUART**, the godmother of the event (CHR)
- `format="$N"` : displays the full name of the related entity if it exists
 - Example : `<prop path="INDI: CHR: XREF" format="{ $N}">` displays **STUART, Marie (° 30 march 1909 + 16 December 2008) (I0298)**
- `format="$v"` : displays the type of related entity followed by the full name of the related entity
 - Example : `<prop path="INDI: CHR: XREF" format="{ $v}">` displays **Godmother : STUART, Marie (° 30 march 1909 + 16 December 2008) (I0298)**
- `format="$V"` : displays the entity number only
 - Example : `<prop path="INDI: CHR: XREF" format="{ $V}">` displays **I0298** if we use the same example as above
 - Example : `<prop path="INDI: BIRT: SOUR" format="{ $V}">` displays **S134**, the ID number of the SOURCE entity supporting the BIRTH event
- `format="text{$e}"` : displays the text "text" if the data exists, nothing otherwise.
 - Example : `<prop path="INDI: NAME: SOUR" format="{ (*) $e}">` displays **(*)** if INDI:NAME:SOUR exists, nothing otherwise
 - Example: `<prop path=INDI: NAME: SPFX max=50><prop path=INDI: NAME: SPFX format="{ $e}"><prop path=INDI: NAME: SURN max=50> <prop path=INDI: NAME: NSFX max=50>` displays surname prefix with a space in between but the space is only displayed if the surname prefix exists. In this example, note that the first 3 `<prop>` statements are plugged together without any space in between, while space can be always included between surname and suffix, because an ending space does not show anyway.
- `max="n"` : limits the display to n% of the available width. Valid for an image or for text.
 - Example : `<prop path="INDI: NAME" max=10">` displays the full name truncated to 10% of the available width
- `length="n"` : limits the display to n characters. Valid for a text.
 - Example : `<prop path="INDI: NAME" length=10">` displays the full name truncated to 10 characters.

<media> tag

```
<media [path="path:to:media"] [max="n"]>
```

Displays the media of the entity. If a path is specified, the corresponding media is used, otherwise any first media found in the entity is displayed.

The "**max**" parameter limits the size in width as above.

Parameters in square brackets are optional.

<name> or <i18n> tag

```
<name|i18n [tag="tag"] [path="path:to:prop"] [entity="tag"] [underline="yes"] >
```

Displays the name of the label given by **tag**, **path** or **entity**.

One of the 3 attributes must be specified. **tag** takes precedence over **path**, which takes precedence over **entity**.

The **underline** parameter can be set to 'yes' or 'no' to underline the text. By default, it is 'no'.

Example:

- `<name tag="NAME" >` displays **Name**
- `<name path="INDI: NAME" >` displays **Name**
- `<name entity="SUBM" >` displays **Submitter**

<mark>

```
<mark path="path:to:prop" [test="value"] [repeat="n"] [default="path:to:other:prop"]  
[notpresent="true"] >
```

Displays a square marker on a blueprint.

The marker is only displayed if the indicated property (**path**) exists in the entity, and if the value of the property is the same as the value of the "**test**" attribute.

The **repeat** attribute is optional and specifies the number of times the marker should be repeated.

The **default** attribute is optional and allow to check this property if the property indicated by **path** doesn't exist.

The **notpresent="true"** parameter allow to display a marker only if the property doesn't exist.

For instance, the following code displays a green marker for individuals whose name is CAPET, and a red marker for individuals whose name is HABSBURG.

```
<font color=green><mark path=INDI: NAME: SURN test=CAPET></font>
```

```
<font color=red><mark path=INDI: NAME: SURN test=HABSBURG></font>
```

This results in the following view in the dynamic tree for instance:



Another example . The following code displays 5 markers for individuals whose sex is unknown.

```
<mark path=INDI: SEX test=Unknown repeat=5>
```

`<mark path=INDI: BIRT: SOUR notpresent=" true">` : Displays a marker if no source is defined for the birth.

User example

Here is an example of marking provided by Sylvain, a user, to put a color marker depending on the source of an event.

```
<td width=0><table>
<font color=#38b020><mark path=INDI: BIRT: SOUR test="Acte de Baptême ( S198)"></font>
<font color=#38b020><mark path=INDI: BIRT: SOUR test="Acte de Naissance ( S199)"></font>

<font color=#3f81d1><mark path=INDI: BIRT: SOUR test="Acte de Baptême collatéral ( S95)"></font>
<font color=#3f81d1><mark path=INDI: BIRT: SOUR test="Acte de Naissance collatéral
( S500)"></font>
<font color=#3f81d1><mark path=INDI: BIRT: SOUR test="INSEE - Fichier des personnes décédées
( S421)"></font>
```

```

<font color=#3f81d1><mark path=INDI: BIRT: SOUR test="Livret de Famille (S2)"></font>

<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé France Naissances/Baptêmes 1546-1896 FamilySearch (S411)"></font>
<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé Geneanet (S126)"></font>
<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé Généalogie en Corrèse (S252)"></font>
<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé Cercle Généalogique d'Alsace (S364)"></font>
<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé Allemagne Naissances/Baptêmes 1558-1898 FamilySearch (S657)"></font>
<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé Généalogie Entraire Recherche Côte d'Or (S657)"></font>
<font color=#7326d1><mark path=INDI: BIRT: SOUR test="Relevé Centre Généalogique de Haute-Marne (S725)"></font>

<font color=#ffa60d><mark path=INDI: BIRT: SOUR></font>

<font color=#ff053b><mark path=INDI: BIRT: DATE></font>
<font color=#ff053b><mark path=INDI: BIRT: PLAC></font>

<font color=black><mark path=INDI: BIRT: SOUR notpresent="true"></font>
</table></td>

```

This code puts a coloured marker for births whose source is filled in, with a different colour depending on the origin of the source, and a black marker otherwise.

Directory where blueprints files are stored

Blueprints are files of HTML codes that are stored in the **User Directory**, in a sub-directory called "**ancestris/blueprints**".

The blueprints directory itself contains several sub-directories, each of them having the title corresponding to the entity to which it applies: it is in each of these sub-directories that the

different blueprints are located.

- **INDI** for blueprints about **individuals**
- **FAM** for blueprints about **families**
- **OBJE** for blueprints about **multimedia objects**
- **NOTE** for blueprints about **notes**
- **SOUR** for blueprints about **sources**
- **SUBM** for blueprints about **submitters**
- **REPO** for blueprints about **repositoris**

As these are HTML format files, you can open them with your favorite internet browser.

Feel free to share your blueprints with the **Ancestris community**! Thanks in advance.

Tutorial

We have written a tutorial in PDF format. It is written in French. Please feel free to download it from **here**.

Revision #29

Created 24 May 2020 16:57:25 by frederic

Updated 23 July 2024 23:12:19 by frederic