November 2014 Features Update

Table of Contents

Illustration Index

Global

The script `common.inc`, which is included at the beginning of every page generating script usually generates debugging output if Debug=Y is passed as a parameter to the script. This created a problem with scripts that generate XML output, rather than HTML. Browsers are implemented to tolerate illegal HTML, such as output generating statements that are not contained in the body section of the page, so this debug output is visible and useful in web pages. XML, however, has strict rules which must be followed, and output that occurs before the top tag is strictly forbidden. To avoid both the failure of XML and XHTML output and avoid the tolerated output of trace information before the body section in HTML, this trace information is moved to the global `$warn` variable, which scripts then have the option of displaying in a legal position within the page to show the trace contents.

Family Tree Enhancements

The nominal index script LegacyIndex.php, which is used to search for individuals in the family tree by name is significantly enhanced to provide more information about the individuals.
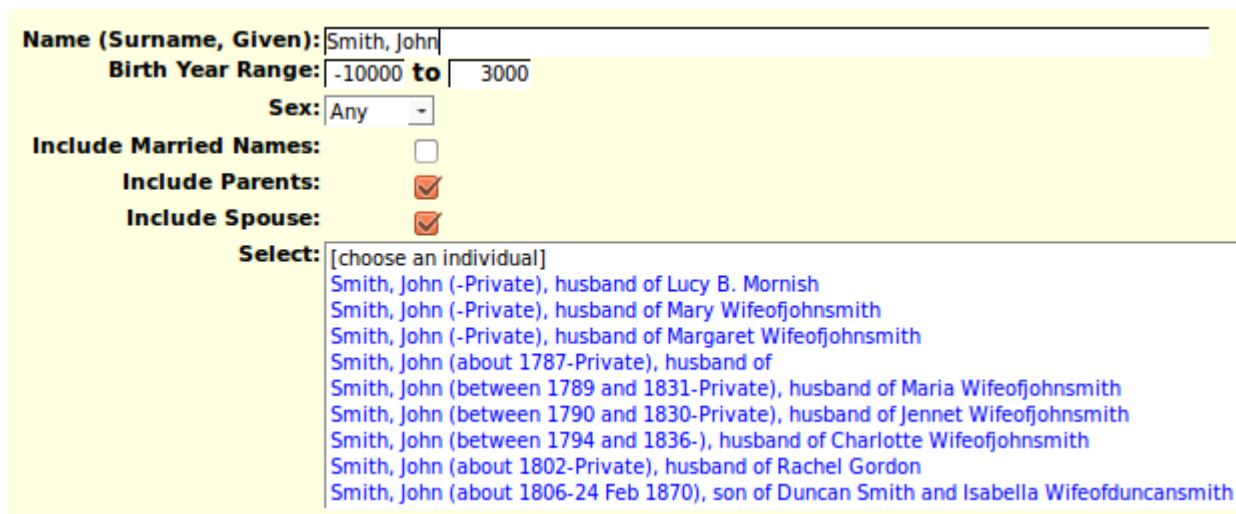
Illustration 1: Additional Information and Options on Nominal Index

There are three new options that you can specify, permitting you to limit the sex of matches, and to control the inclusion of information about the parents and spouse of each individual. In many cases this will eliminate the need to examine multiple individuals with similar names and birth dates to identify the specific individual you are interested in. As the family tree continues to grow this will become more and more valuable.

As discussed last month while most events are implemented by rows within the table of events, through instances of LegacyEvent, some events were formerly represented by sets of fields within either the LegacyIndiv or LegacyFamily records. Migrating these events to the common table of events exposed some issues which were not completely addressed by the changes made last month. In the old implementation birth and christening events were presented prior to general events because of the implementation, while death and burial events were presented after all general events. With all events migrated to the common table of events the order of presentation is based upon a numeric order field, which is manipulated by the button to sort events by date. The migration of events takes place gradually as individuals in the family tree are modified, not all at once. This revealed a bug where a single event might momentarily have citations associated with its new location in the common table of events, as well as with its old location. All citations are now associated to the new location at the same time that the event is moved. The implementation of access to the contents of an instance of LegacyIndiv was also changed this month to force application code to access the event information only through its new location.

A new method LegacyFamily::getMarEvent is added to parallel the LegacyIndiv::getXxxxEvent methods, in preparation for moving the marriage event to the common event table. An option is added to the method LegacyFamily::getName to omit the date of marriage. This is used by the LegacyIndex.php script as described above.

The method `LegacyName::getNames`, which returns an associative array of instances of `LegacyName` matching a set of field values, creates instances of `LegacyName` which have one extra field, which is not in the associated table. This field is the gender field from the associated instance of `LegacyIndiv`. This facilitates the use of this method by the script `addCitXml.php`.

The scripts `EditFamilies.php` and `EditParents.php`, which permit updating information about families are changed to initialize the display using PHP instead of using AJAX to retrieve information from the database which is then used to populate the page. This eliminates a sometimes annoying delay while the page was completed. The "Edit" family button and the "Add Family" button now refresh the page with the new options instead of using AJAX. The net result is that the application seems more responsive.

In the scripts `EditFamilies.php` and `EditParents.php` when the birth date of a child is changed the associated internal sort value is also updated. As a result when the user clicks on the "Order Children by Birth Date" field the children will be sorted by the updated birth dates, not by the values from the database records that were read for the original display. This is particularly important for children that have been added to the display but not yet written to the database, which does not happen until the family is saved, or the individual child is edited and then saved. So you can now enter an entire family, for example, adding all the children to the display, entering their birth dates, and then click on "Order Children by Birth Date" and it will function as expected.

In the scripts `EditFamilies.php` and `EditParents.php` the button to "Order Events by Date" was not previously implemented.

When the script `editEvent.php` is used to add or update an occupation event the description row is modified to identify the purpose of the field as holding the actual occupation, an appropriate set of abbreviation expansions is associated with the field, and the popup help describes the use of the field to hold an occupation description.

There are several improvements to the linking between citations in the family tree and vital statistics and census records.

The default child window in which `editMarriages.php` and `editParents.php` are opened is made wider so that all of the buttons can be seen.

Census Database Enhancements

Adding a census citation for a birth registration now updates the 1851, 1861, and 1921 censuses to define a link to the individual.

Vital Statistics Enhancements

The site now supports transcribing vital statistics for any province or state. For the United States of America this means the page listing all of the states and their associated postal abbreviation now also has a link for each state to the special resources for each state. Each state and province page now has a link to the list of Counties for that state or province. For example the list of counties for Michigan looks like:

United States of America: Michigan: County Table

You are not authorized. **Sign on** to update the database.

ID	Name	Created	Ended	Delete	Townships
001	Alcona	1852	9999	Delete	Townships
003	Alger	1852	9999	Delete	Townships
005	Allegan	1852	9999	Delete	Townships
007	Alpena	1852	9999	Delete	Townships
009	Antrim	1852	9999	Delete	Townships
011	Arenac	1852	9999	Delete	Townships
013	Baraga	1852	9999	Delete	Townships

Illustration 2: List of Counties for Michigan

This part of the database is extremely incomplete. Before vital statistics for an area can be transcribed on the site it is necessary to define the county, and the townships within the county for which vital statistics are to be recorded.

The functionality of a County is now encapsulated in a PHP class.

When viewing an individual birth, death, or marriage registration transcription a link is added in the page header and footer to the summary of transcriptions for the current year and township.

Bug Fixes

- Method `LegacyIndiv::getFamilies` performs a fixup for an invalid `idmrpref` value, that is a value that does not refer to any instance of `LegacyFamily` associated with the individual, if the current user is authorized to update the individual.

- If children were added to an instance of `LegacyFamily` before the family was saved there was a failure because the identification value IDMR was not available. The implementation is changed to update the IDMR value in the instance of `LegacyChild` when the instance of `LegacyFamily` is saved.

- The `LegacyCitation` constructor used the incorrect value of the associated record key when determining the value of the citation order field to add a new citation after all of the existing citations to a fact or event.

- In PHP class `County` the county identification code field is always stored as a string even if its value is numeric. This is to support US states where counties are internally represented by numeric statistical identifiers.

- During the deployment of the new event support there was a period where it was possible to create multiple preferred events of a particular type. Code is added to the script `editIndivid.php` to correct this when it is discovered.

- The "init" parameters passed to the `editIndivid.php` script did not update the new event locations.

- Handling of an invalid IDIR parameter value is improved in `editIndivid.php`.

- The script `editEvent.php` constructed an invalid name for the input field in a citation row that contains the numeric key of the associated master source record.

- The script `editEvent.php` did not display the full list of citations for an event that was in the process of being migrated to the new table. This is corrected by using the method `LegacyEvent::getCitations`.

- There was a crash in `editEvent.php` trying to add a citation to an event which had an unresolved location name. The location name is resolved first.

- The addition of birth citations to the 1851, 1861, and 1921 censuses of Canada did not cause the indicated page to be searched for a matching individual, and a link back to the individual in the family tree set.