

August 2014 Features Update

Table of Contents

Global.....	3
Family Tree Enhancements.....	6
Census Database Enhancements.....	8
Vital Statistics Enhancements.....	9
Bug Fixes.....	10

Illustration Index

Illustration 1: Sign In Dialog with Option to Remember Userid and Password.....	4
Illustration 2: Account Management Dialog with Messages.....	5

Global

The process for registering a new contributor is now completely automated. When a new user completes the registration dialog an e-mail is sent immediately by the site. When the new user receives this e-mail and clicks on the contained confirmation link this verifies that the supplied e-mail address is valid, which is all that is required to confirm the new registration. The new user is sent a welcome message with further instructions.

The registration page now accepts a user name that contains an @ (at) sign. This is a convenience in that it simplifies the registration process but is not recommended because it exposes the user's e-mail address.

As access to the database is further encapsulated by class methods, static `get`, `update`, and `delete` methods are being added to each class. These are used to apply an operation to a set of instances of the class that match characteristics specified by a list of field names and values to match. In addition to field name values that are used to restrict the set of records that the operation applies to, the `get` method also supports specifying the beginning “offset” and “limit” of number of records returned so that a large set of matching records can be retrieved a block at a time. This supports those web pages that permit browsing through the set of records matching a request. The total number of records that would have been returned if a limit had not been specified is always returned by modifying the parameter array to add an element named “count”. If the limit on the number of records returned is set to zero in the parameters the `get` method returns null.

In particular all access to the table of source citations, `tblSX`, is now encapsulated by the class `LegacyCitation`. This class is used to view, add, delete, update, and manage citations in all scripts. This is a global feature because vital statistics and census record scripts can add citations to individuals in the family tree. The class now has static methods `updateCitations`, and `deleteCitations` which permit applying changes to all citations that match certain requirements. In particular these methods provide the ability to change or delete all citations to facts or events recorded in specific records in the family tree database. The class now overrides the method `toXML` so the resulting XML document includes interpretations of key fields.

The class `LegacySource` has added methods `getSources`, `updateSources`, and `deleteSources` to encapsulate all access to the table of master sources.

The classes `LegacyEvent` has implemented the static `getEvents` and `deleteEvents` methods, but not the `updateEvents` method. The class `LegacyFamily` has implemented the static `getFamilies` and `deleteFamilies` methods, but not the `updateFamilies` method. The class `LegacyPicture` has implemented the static `getPictures` and `deletePictures` methods, but not the `updatePictures` method. The class `User` has implemented the static `getUsers` method, but not the `update` and `delete` methods.

To encapsulate the function of merging two instances of `LegacyAddress` in the database, so the script code does not have to directly update the database, a new static method `mergeAddresses` is added to the class `LegacyAddress`.

When signing on the user is now presented with an option to remember the user name and password on the current computer. If selected every time you enter the site subsequently you are automatically logged on. This option should, of course, never be selected on a shared computer as it means that there is a file on the computer that contains your user name and password as text.

Sign In for Contributing [? Help](#)

You can contribute to this web-site by adding transcriptions of census pages, birth/marriage/death registrations, or by adding information about your family to the family tree. If you are not already a subscriber to this web-site, please click on the **Register** button or hold down the Alt shift and press R.

In compliance with European Union regulations please note that this site uses a Cookie to record the fact that you are signed on.

User Identifier:

Password:

Remember: **Remember me on this computer.**

Illustration 1: Sign In Dialog with Option to Remember Userid and Password

A consequence of the automatic signon is that the user is not presented with any messages that have been sent by other contributors, as that is a side effect of the manual signon. The account management dialog, accessed by clicking on the button at the top right of every page, which displays your user name, is enhanced to display any pending messages as well.

Account Management [? Help](#)

User Identifier:

Current Password:

New Password:

Repeat New Password:

E-Mail Address:

Send Notices to E-mail:

Suppress Help Popups:

Messages:

At 2014-08-31 12:36:35 user 'jacobban' wrote:

About: [Family Tree] UserInfo.php

Demonstration

Sender's e-mail address: webmaster@jamescobban.net

Illustration 2: Account Management Dialog with Messages

Family Tree Enhancements

As also discussed above there are significant improvements to the separation of functionality presented to the user from the internal implementation of the database.

Although the original and principal use of the class `LegacyEvent` is to represent event or fact records contained in the events table, `tblER`, it is also used to present an object-oriented view of events that are actually represented in the current implementation by sets of fields in other tables. For example the primary birth event for an individual is represented by a set of fields in the main individual record in `tblIR` in order to simplify and speed up searches for individuals by birth date. To avoid having the actual site logic be dependent upon this internal implementation, the primary birth event, primary death event, primary burial event, primary marriage event, and sacraments of the Church of Latter Day Saints are now accessed through special instances of `LegacyEvent` which mask the internal implementation.

The script `deleteMarriage.php` is renamed to `deleteMarriageXml.php` to make it clear that it returns an XML document, not an HTML page, and is intended to be invoked by Javascript code through AJAX.

The script `showAddedIndividuals.php`, which is used to display the individuals added in a specific week, starting with the week ending today, is changed to use the date the record was added to the table of individuals, rather than looking for INSERT records in the log file. The information about each individual now includes the name of their spouse(s). The title of the page now displays the names of the months rather than the month number.

The script `getSourcesXml.php`, which is used by Javascript code to obtain a list of sources for the construction of `<select>` HTML form elements, particularly when adding citations, is enhanced to use `LegacySource::getSources` and `LegacySource::toXml` to obtain the information to send. Additional parameters are supported to the script to permit selecting sources depending upon the values of any field, and to sort the response by any field or set of fields.

The script `LegacyAddress.php`, which is used to display or edit an entry in the table of addresses, is enhanced to support the use by `LegacyAddress.js` of the script `mergeAddressesXml.php`. HTML layout and the PHP code for setting up the values of input fields is separated for easier maintenance. The debug flag is passed to the next script. If the debug flag is set then `mergeAddressesXml.php` is invoked by submit rather than through Ajax, so the user can see the contents of the XML document that is returned.

The script `mergeAddressesXml.php` is changed to use the method `LegacyAddress::mergeAddresses` rather than updating the database using SQL. The parameter validation of this script is improved.

The script `DeleteSourceXml.php`, which is used by the page `LegacySources.php` to delete an individual Master Source record, is changed to use the method `LegacySource::delete` to delete the source.

The page `LegacySources.php` is changed to use method `LegacySource::getSources` to obtain the set of sources requested by the user and the method `LegacyCitation::getCitations` to determine how many citations there are to a particular

source. Parameter validation is improved.

The page `grantUser.php`, which is invoked when an owner of an individual in the database grants shared ownership to another user, is changed to use the method `LegacyIndiv::getFamilies` to get the set of families in which an individual is a spouse, the method `LegacyIndiv::getParents` to get the set of families in which an individual is a child, and the method `LegacyFamily::getChildren` to get the set of children of a family, rather than using SQL to extract information from the family tree, as the script traverses the tree to identify all the ancestors and descendants of the initial individual.

The script `mergeUpdIndivid.php` is changed to use the method `LegacyCitation.updateCitations` to redirect existing citations to the second individual, who is being merged into the first individual, so they refer to the merged individual.

The dialog `chooseIndivid.php` is enhanced to use a CSS formatted popup to display warning messages rather than the browser alert function.

The script `getRecordXml.php`, which is used to obtain an XML document containing the contents of any record in the family tree no longer requires that the user be signed on to run, although some information is suppressed.

If the current user is not signed on, or is not an owner of an individual whose information is private, the page `legacyIndivid.php` now explains the process for gaining access to the private information.

Census Database Enhancements

When searching for citations from the family tree to a census page the script `matchCitations.php` now supports extra spaces in front of the page number in the citation text to permit citations that sort in natural order.

Different portions of the census database code formerly updated two different cookies to record the same information: the last census district referenced for each census year. Now a single cookie is updated so the behavior is more predictable.

Vital Statistics Enhancements

When updating a birth registration the script `BirthRegUpdate.php`, which applies the changes to a birth registration that were entered on the page `BirthRegDetail.php`, now uses the method `LegacyCitation::getCitations` to obtain the list of matching citations. The associated individual is examined. If the current date of birth is blank or an approximation then it is replaced with the date from the birth registration. If the location of birth is blank it is replaced with the birth location from the birth registration. Among other things this ensures that the citation to the birth registration which is added to the birth event is visible when the individual is viewed. A citation is also added for the name of the individual. This citation contains the actual name recorded in the birth registration as citation text.

When updating a death registration the script `DeathRegUpdate.php`, which applies the changes to a birth registration that were entered on the page `DeathRegDetail.php`, now uses the method `LegacyCitation::getCitations` to obtain the list of matching citations. The associated individual is examined. If the current date of death is blank or an approximation then it is replaced with the date of death from the death registration. If the location of death is blank it is replaced with the death location from the birth registration. Among other things this ensures that the citation to the death registration which is added to the death event is visible when the individual is viewed. A citation is also added for the cause of death of the individual, which is also set if it was previously blank.

The summary display of birth registrations, produced by `BirthRegResponse.php` in response to a query from `BirthRegQuery.html`, now contains a “Delete” button if there is an actual record in the database. This is most commonly used to remove a record which has the wrong registration year or registration number, since the site does not currently provide a mechanism for renumbering a registration.

The summary display of death registrations, produced by `DeathRegResponse.php` in response to a query from `DeathRegQuery.html`, now contains a “Delete” button if there is an actual record in the database. This is most commonly used to remove a record which has the wrong registration year or registration number, since the site does not currently provide a mechanism for renumbering a registration.

For consistency with other scripts the parameter to limit the maximum number of rows in a response to a birth or death registration query which is passed to `BirthRegResponse.php` or `DeathRegResponse.php` is renamed to “Limit” from “Count” to more clearly identify its purpose.

A new script `deleteBirthRegXml.php` is added to permit the code executed in the browser to implement the dynamic function of the page `BirthRegResponse.php` to request the server to delete an instance of a Birth registration record from the database.

A new script `deleteDeathRegXml.php` is added to permit the code executed in the browser to implement the dynamic function of the page `DeathRegResponse.php` to request the server to delete an instance of a Death registration record from the database.

Bug Fixes

- Selecting the option to suppress context specific popup help set the e-mail option flag by mistake.
- It was possible to turn the e-mail and popup help suppression options, but not to turn them off.
- The initialization of the session cookie containing information about the options selected by the user is moved to be performed in `Signon.js` instead of `UserInfo.php` because with the automatic logon feature `UserInfo.php` is not always invoked during a logon. Also it is better to set all cookies from Javascript running on the client's browser rather than by commands sent from the server.
- The method `LegacyFamily::save` emitted some XML tags even when the XML option was false. This resulted in unexpected text appearing before the beginning of the web page.
- The method `LegacyEvent::save` is corrected to save the associated record in the case where the event does not represent a row from the table `tblER`.
- The method `LegacyEvent::setDate` did not correctly update the date fields in the associated record in the case where the event does not represent a row from the table `tblER`.
- An exception was thrown if the string passed to `LegacyEvent::setLocation` did not match any existing location record. This failure occurred if the invoking script did not validate the location name before calling `LegacyEvent::setLocation`. However new features to add information to the family tree as a side-effect of linking a vital statistic or census record line to an individual in the family tree do not have the ability to interact with the user to resolve this validation.
- The PHP function `htmlentities` was misspelled on one line in the class `Record`.
- The methods of class `LegacyIndiv` for obtaining instances of the class `LegacyEvent` that represent events that are actually recorded by sets of fields in the individual record in table `tblIR` did not pass the pointer to the instance of `LegacyIndiv` to the constructor for the new instance of `LegacyEvent`, resulting in a script failure when the instance of `LegacyEvent` tried to update the main record.
- Citations to pages in a census do not sort in the expected page order. For example citations to page 1 are followed by citations to pages 10 through 19, followed by citations to page 2, and so on. It is suggested that when citing to pages 1 through 9 of a census enumeration division that you insert an extra space before the page number, so the strings sort as expected. Supporting the extra space in the citation text required a change to the patterns used in class `LegacyIndiv` to match census citations.
- The class `User` is corrected to not construct a new instance if it is only passed an e-mail address.
- The common library routine `logSqlUpdate` no longer generates debug output. That is the responsibility of the invoking script.

- The common census query handling module `QueryCensus.php` no longer requests the browser to create a cookie to record the last census sub-district selected. That functionality is moved to Javascript code running in the browser.
- In `LegacyAddress.js` the name of the button for merging addresses was misspelled.
- The script `relationshipCalculator.php` displayed the wrong birth and death dates for the second individual.
- The script `pageUpdate.php`, which performs the actual update of the census database for changes made in the detailed census page form, incorrectly issued a warning message for page numbers greater than 99.
- The script `MarriageRegResponse.php` was too restrictive in the range it permitted for the birth year of a participant. It now supports the range 1790 to 2000.
- An exception was incorrectly thrown if the link to an individual in the family tree from a death registration was cleared.