

# February 2018 Features Update

## Table of Contents

Global.....	3
Templates.....	3
Enhancements to Common Record Support.....	4
Support for Sets of Records.....	6
Support for Geographical Entities.....	7
Videos.....	13
Style Classes.....	13
Surnames.....	14
Given Names.....	15
Family Tree Enhancements.....	16
Census Database Enhancements.....	17
Vital Statistics Enhancements.....	18
Bug Fixes.....	19

## Illustration Index

Illustration 1: Example of output of \$record->dump showing identification of fields depending upon their role.....	6
Illustration 2: Country Table.....	7
Illustration 3: The United States of America in Multiple Languages.....	8
Illustration 4: Management Page for States/Provinces within a Federal State.....	9
Illustration 5: Managing Counties in Ontario.....	10
Illustration 6: Form for Managing the Towns and Townships for which there were Division Registrars of Vital Statistics.....	11
Illustration 7: The Form for Managing Languages.....	12
Illustration 8: Sorry Canada does not support Spanish.....	12
Illustration 9: Manage Tutorial Videos.....	13

## Global

The sudden death of my wife interrupted the focus I required to maintain this log of changes, so this report covers almost two years of changes to the services and implementation.

## Templates

The major implementation change is in more completely following the Model-View-Controller (MVC) design model. In the context of this site:

- The Model is the database of family tree and document transcriptions and the mechanism by which that information is organized for use by applications.
- The View is the web page layouts which make the information on the Web server accessible to users.
- The Controller is the interactive features of the web pages which are performed within the client's browser.

In the original implementation there was not a clear separation of the Model and the View. The same web server based scripts both extracted information from the database and created the HTML pages which are sent to the browser. That functionality is now clearly split: the application scripts running on the server no longer contain any code for laying out HTML, instead the application scripts use a template engine which takes logically structured information and inserts the values into specific points in a template.

At the moment you can see the templates by pointing your browser at <https://www.jamescobban.net/templates/>. Eventually the contents of that directory will be hidden because there is nothing that can be directly meaningful to clients of the database. This directory is only normally accessed by the application logic running on the web-server, not by the browser, however it will help you to understand the change in design paradigm if you look at some of these templates in your browser. As you can see these are just partial HTML pages which your browser will display visually.

A particularly important template is the [master page template](#). This template controls the appearance of the common portion of all pages. When an application script prepares a page for presentation to your browser it supplies substitution values for each of the variables identified within the template by being preceded by a dollar sign (\$), most importantly the variable \$MAIN is where the entire body of the page is inserted. In the original implementation the presentation information contained in this template was generated in dozens of different files on the server. In the new implementation the appearance of the common portions of all of the pages is derived only from this single template<sup>1</sup>. That means that if the site management decided to change the appearance of the header and footer of all pages on the site only this single template file needs to be changed.

1 Actually this template has a partner which handles the presentation for those clients who are still running Windows XP with Internet Explorer 7 because IE7 does not support HTML5. If you are among them *please* upgrade. Windows XP is a security nightmare because the hacker-verse knows *every* weakness in that system and Microsoft stopped shipping security fixes years ago. There is also a `template/dialog.html` which handles pages which are displayed only as subordinate pages of a main editing dialog, for example the page for adding a citation to an event. To avoid clutter and minimize scrolling these subordinate pages do not have a header section.

Similarly the contents of the main section of each page comes from another template file. Each application script on the server typically selects from a set of templates based upon information about and from the client. For example the main page for each supported country is generated by the application script `/FamilyTree/genCountry.php`, based upon the parameters: `CC` which identifies the country by its ISO country code, and `lang` which identifies the preferred language of presentation of the client by its ISO identifier. The application script selects one of the template files `"genCountryccll.html"` where `"cc"` is the ISO country code and `"ll"` is the ISO language code. As a result adding support for another country or another client language is reduced to simply creating a new template file.

Once you have selected a preferred presentation language that information is passed between scripts by a `lang=` parameter. Not all pages support every possible language. There are 182 languages recognized by the application. At the moment many parts of the system have been translated into French, and a few pages into German and Spanish.

## ***Enhancements to Common Record Support***

There have been a lot of improvements to the common handling of database records:

- Change method `toXML` in all database record classes to always accept a single options parameter and always return the XML and only print it by default.
- Permit a derived class to insert a record with a known primary numeric key if the class has identified that the record is improperly missing.
- Assume a default behavior for a table with no defined characteristics. In particular if the table does not have a defined set of fields with default values and data types, the contents of the very first record in the table is used in its place.
- Rename method `getField` to `get` and method `setField` to `set` to improve the readability of application code.
- Add method `toJson` parallel to `toXML`.
- Add support for new tables ``Countries``, ``CountryNames``, ``Languages``, ``Domains``, ``Nicknames``, ``FieldComments``, ``Concessions``, and ``Videos``.
- Do not throw an exception if method `save` is called by a client who is not authorized to update the database. Just ignore the request.
- Do not throw an exception if there is no record in the database corresponding to the identification fields passed when creating the an instance. If there is no matching record in the database `$record->isExisting()` returns false.
- Do not throw an exception if the field requested by either `$record->get($name)` or `$record[$name]` does not exist. Instead display a warning message to the user, log the error for the administrator, and return null.
- Do not throw an exception if the field requested by either `$record->set($name, $value)` or `$record[$name] = $value;` does not already exist. These operations now define the value of a a temporary field in the record. This temporary field can be accessed like the existing permanent fields either by methods `get` and `set` or by subscripts but their values are

discarded when the record is saved to the database. These extra fields are used to support using the template engine to lay out the contents of a portion of a database table.

- The table `SqlLog` which is used to track every change made to the database is changed to record before and after images of each record changed by the request. This will permit backing out changes to the database. The associated global function `logSqlUpdate` is changed to support the new parameters and is deprecated in favor of a new method `$record->log`.
- Adding support for the new `SqlLog` created the opportunity to implement a standard mechanism which permits the application to allow the client to change fields in the primary key of a record when saving the record back to the database. In essence an application can now *move* a record from one part of the database table to another location.
- All of the classes representing records from the family tree which formerly included "Legacy" as part of the name are renamed to remove that prefix. For example class `LegacyCitation` is renamed to class `Citation`. This improves the readability of the application code and reflects the fact that although some portions of the database definition were derived from the database implemented by the Legacy Family Tree application on Windows, none of the code came from that product and the database definitions have substantially deviated from the original over the last fifteen years. Internal references to the database documentation obtained from Millenium Corporation maintain the copyright notices.
- Surname matching patterns were defined in an internal table in the `Record` class definition. These have been moved to the enhanced `Surnames` table where they can be maintained more easily by the administrator and are visible as part of the `Surname` object.
- The `Record` class implements the interfaces `Iterator` and `ArrayAccess` which permit the application code to access the contents of a record by using either `foreach` or by subscripts as if it was an array.
- The translation between user-friendly external table names and the internal names used by the SQL server was moved into the class `Record`. All tables whose internal name is not an English description of the contents is assigned a meaningful external name which is a plural noun.
- Access by application code and derived classes to the internal implementation of class `Record` is limited by changing the access mode of internal fields and in their place defining new techniques such as method `Record::getInformation($tableName)` which consolidates read-only access to the definition of a table, and `static::getRealFieldName($fieldname)` has consolidates standard translation of field names from external to internal names.
- Improve method `$record->postUpdate` including support for field name synonyms.
- Method `$record->getId` is extended to support primary keys that are not `AUTOINCREMENT` integers and to report problems by returning null rather than throwing an exception.
- All SQL read, update, delete, and inserts are now protected against hackers by using prepared statements which separate the definition of the structure of the request from the specific values used with the request. Since this would have made the debugging output generated for errors harder to read add a global function `debugPrepQuery` which takes the prepared statement and the parameters passed to it and constructs a string which represents what the command would have been if prepared statements had not been used.

- Improve support for tables without a table specific class implementation. In particular it is now possible to create new records within such a table.
- The method dump now uses highlighting and different styles of quote marks to distinguish between those fields in a record that are part of the unique key of the record, ordinary data fields from the database, and extra fields set to support the template engine. In this output the unique key fields of the record are in italics, the permanent fields are enclosed in back-ticks (AKA left single quotes), and temporary fields are enclosed in vertical apostrophes.

test	
<i>'filename'</i>	<b>TutorialIndividual</b>
<i>'lang'</i>	en
<i>'description'</i>	<b>a Tutorial on the main page for an Individual</b>
<i>'display'</i>	Y
<i>'row'</i>	1
<i>'even'</i>	odd

*Illustration 1: Example of output of \$record->dump showing identification of fields depending upon their role*

- add default support for tables with multiple primary keys

## Support for Sets of Records

A new class **RecordSet** is defined as a standard representation of a portion of a table. This replaces a lot of code throughout the system where individual applications created an array of records from a table, deleted a subset of records from a table, or updated a set of records. Formerly many classes defined static methods for performing these functions, which is a sub-optimal design. Ordinary class methods could not be used because these functions did not act on an individual record. Defining the class RecordSet permits sharing more code between classes. Most classes derived from RecordSet only redefine the construction of the WHERE clause and do so exploiting common tools provided by the base implementation. Furthermore the base class implementation can be used for many classes that do not have special requirements for parameter validation. The template engine supports RecordSet as if it was an array of arrays. An example of its use is:

```
$parms      = array('limit'      => $limit,
                  'offset'     => $offset,
                  'order'      => 'SrcName');
if (strlen($pattern) > 0)
    $parms['SrcName'] = $pattern;
$sources    = new RecordSet('Sources', $parms);
$numRecs    = $sources->count(); // number in the set
$info       = $sources->getInformation();
$count      = $info['count']; // if "limit" omitted
$query      = $info['query']; // SQL SELECT generated
```

```

foreach($sources as $idsr => $source)
{
    // act on each individual $source
    ...
}
// act on each individual $source
$template->updateTag('source$idsr', $sources); // template insert

```

All displays of the contents of tables now support restricting the number of rows displayed and to browse through the table one page at a time, with the default page size being 20 rows.

## Support for Geographical Entities

A new table Countries has been added and is managed by a class Country which represents the behavior of an individual country. This table contains information about every country recognized by the United Nations including the English language name of the country. This table is viewed and managed through the [Countries Management Page](#).

## Country Table Update [? Help](#)

returned rows 1 to 20 of 242 [-->](#)

Pattern:

ID	Name	Other Names	States	Genealogy	Delete
AF	Afghanistan	Names	States	<a href="#">Afghanistan Genealogy</a>	Delete
AX	Aland Islands	Names	States	<a href="#">Aland Islands Genealogy</a>	Delete
AL	Albania	Names	States	<a href="#">Albania Genealogy</a>	Delete
DZ	Algeria	Names	States	<a href="#">Algeria Genealogy</a>	Delete
AS	American Samoa	Names	States	<a href="#">American Samoa Genealogy</a>	Delete
AD	Andorra	Names	States	<a href="#">Andorra Genealogy</a>	Delete
AO	Angola	Names	States	<a href="#">Angola Genealogy</a>	Delete
AI	Anguilla	Names	States	<a href="#">Anguilla Genealogy</a>	Delete
AG	Antigua and Barbuda	Names	States	<a href="#">Antigua and Barbuda Genealogy</a>	Delete
AR	Argentina	Names	States	<a href="#">Argentina Genealogy</a>	Delete
AM	Armenia	Names	States	<a href="#">Armenia Genealogy</a>	Delete
AW	Aruba	Names	States	<a href="#">Aruba Genealogy</a>	Delete
AU	Australia	Names	States	<a href="#">Australia Genealogy</a>	Delete
AT	Austria	Names	States	<a href="#">Austria Genealogy</a>	Delete
AZ	Azerbaijan	Names	States	<a href="#">Azerbaijan Genealogy</a>	Delete
BS	Bahamas	Names	States	<a href="#">Bahamas Genealogy</a>	Delete
BH	Bahrain	Names	States	<a href="#">Bahrain Genealogy</a>	Delete
BD	Bangladesh	Names	States	<a href="#">Bangladesh Genealogy</a>	Delete
BB	Barbados	Names	States	<a href="#">Barbados Genealogy</a>	Delete
BY	Belarus	Names	States	<a href="#">Belarus Genealogy</a>	Delete

Update Database
Add New Country

Illustration 2: Country Table

Most registered users have permission to update this table although I trust you to use that privilege wisely. The main table has been completely populated with all 238 countries recognized by the United Nations. This page is a portal to two extensions:

- You can choose to supply the spelling of the name of a country in another language by clicking on the "Names" button. Casual visitors can see the information but cannot modify it. For example the name of the United States of America is at this moment defined for four languages:

**United States of America: Names Table Update** [? Help](#)

Language	Name	Article	Possessive	Delete
es	Estados Unidos del America	los	dos	Delete
fr	Etats Unies de l'Amerique	les	des	Delete
en	United States of America		of	Delete
de	Vereinigte Staaten von Amerika	die	von die	Delete

Update Database    Add New Language Name

*Illustration 3: The United States of America in Multiple Languages*

- As a registered contributor you can supply or manage the names of component states or provinces of a federal state by clicking on the "States" button. Each "State" is represented in the database by a record in the "Domains" table. The following shows the table of states and territories in the United States of America as currently defined based upon the US Postal Service addressing table. Casual visitors can see this table but cannot make any changes.



# United States of America: State Table Update

[? Help](#)

showing rows 0 to 20 of 55

Country:

Language:

[-->](#)

Code	Name	Delete	Counties	Resources
USAL	Alabama	Delete	Counties	Details
USAK	Alaska	Delete	Counties	Details
USAS	American Samoa	Delete	Counties	Details
USAZ	Arizona	Delete	Counties	Details
USAR	Arkansas	Delete	Counties	Details
USCA	California	Delete	Counties	Details
USCO	Colorado	Delete	Counties	Details
USCT	Connecticut	Delete	Counties	Details
USDE	Delaware	Delete	Counties	Details
USDC	District Of Columbia	Delete	Counties	Details
USFL	Florida	Delete	Counties	Details
USGA	Georgia	Delete	Counties	Details
USGU	Guam	Delete	Counties	Details
USHI	Hawaii	Delete	Counties	Details
USID	Idaho	Delete	Counties	Details
USIL	Illinois	Delete	Counties	Details
USIN	Indiana	Delete	Counties	Details
USIA	Iowa	Delete	Counties	Details
USKS	Kansas	Delete	Counties	Details
USKY	Kentucky	Delete	Counties	Details

### Illustration 4: Management Page for States/Provinces within a Federal State

Clicking on the "Details" button displays a page of genealogical resources specific to the state, which is customized by the genProvinceccssl.html template where "cc" is the country code, "ss" is the 2 or 3 character unique identifier of the state, province, or territory, preferably as defined by the federal state, and ll is the language of display so the information can be presented in the language of the viewer.

Clicking on the "Counties" displays a page of information about lower level administration regimes as used for recording vital statistics. In Ontario and Michigan, for example, these are counties, in Manitoba they are urban and rural municipalities, and in England they are registry offices as organized for the convenience of the General Records Office and which do not correspond to the civil administration districts or shires. This table is only sparsely filled in as research requires it. For example for Ontario, Canada you will see:

# Canada: Ontario: County Table Update

[? Help](#)

returned rows 1 to 20 of 53

[-->](#)

ID	Name	Created	Ended	Delete	Townships	Location
Alg	Algoma District	1852	9999	Delete	Townships	Location
Brt	Brant	1852	9999	Delete	Townships	Location
Bru	Bruce	1852	9999	Delete	Townships	Location
Cr1	Carleton	1852	9999	Delete	Townships	Location
Dfn	Dufferin	1852	9999	Delete	Townships	Location
Dnd	Dundas	1852	9999	Delete	Townships	Location
Drm	Durham	1852	9999	Delete	Townships	Location
Elg	Elgin	1852	9999	Delete	Townships	Location
Esx	Essex	1852	9999	Delete	Townships	Location
Frn	Frontenac	1852	9999	Delete	Townships	Location
Ggy	Glengarry	1852	9999	Delete	Townships	Location
Gvl	Grenville	1852	9999	Delete	Townships	Location
Gry	Grey	1852	9999	Delete	Townships	Location
Hdm	Haldimand	1852	9999	Delete	Townships	Location
Hbn	Haliburton	1852	9999	Delete	Townships	Location
Htn	Halton	1852	9999	Delete	Townships	Location
Hst	Hastings	1852	9999	Delete	Townships	Location
Hrn	Huron	1852	9999	Delete	Townships	Location
Knr	Kenora District	1852	9999	Delete	Townships	Location
Knt	Kent	1852	9999	Delete	Townships	Location
ID	Name	Created	Ended	Delete	Townships	Location
<a href="#">Update Database</a>		<a href="#">Add New County</a>				

*Illustration 5: Managing Counties in Ontario*

This table is well populated with respect to the post-Durham Report administration of Ontario, which is why all of the counties are currently labeled as starting in 1852. Clicking on the "Location" button takes you to the Location record named "countyname Co, ON, CA" which is the standard name to be used in family trees when identifying that you only know which county an event took place in. As a registered contributor you can update the description of the location entry for the county, for example by adjusting the Google® maps description or adding genealogically significant information about the county. Clicking on the "Townships" button takes you to a page for managing the list of lower level registration administrations for vital statistics. This list should include every township and town in the county for which there was a division registrar for vital statistics. If you have permission to contribute transcriptions of vital statistics then you have the authority to add any missing towns or townships to this list.

## Canada: Ontario: County of Elgin: Township Table

[? Help](#)

ID	Name	Delete	Concessions
Aldborough	Aldborough	Delete	Concessions
Aylmer	Aylmer Village	Delete	Concessions
Bayham	Bayham	Delete	Concessions
Dorchester South	Dorchester South	Delete	Concessions
Dunwich	Dunwich	Delete	Concessions
Dutton	Dutton Village	Delete	Concessions
Malahide	Malahide	Delete	Concessions
Port Stanley	Port Stanley Village	Delete	Concessions
Southwold	Southwold	Delete	Concessions
Springfield	Springfield Village	Delete	Concessions
St Thomas	St. Thomas City	Delete	Concessions
Vienna	Vienna Village	Delete	Concessions
West Lorne	West Lorne Village	Delete	Concessions
Yarmouth	Yarmouth	Delete	Concessions

*Illustration 6: Form for Managing the Towns and Townships for which there were Division Registrars of Vital Statistics*

In this table the first column represents the way that the town or township is recorded in the transcription of each vital statistic, while the Name column shows how it is displayed in the selection menu for greater clarity for the transcriber. All of these fields can be modified by a registered user and the changes are applied if you either click on the Update Database button or press the Enter key while in one of the input fields. The final column is part of a feature that is still under construction which is intended to assist the site to identify more closely the location of a property identified by lot number and concession.

There is also a table of Languages which exists to populate selection lists in places like the alternate country names table, and interpret user-selected presentation languages.

# Language Table Update

[? Help](#)

<--

returned rows 21 to 40 of 182

-->

Pattern:

ID	Name	Native Name	Article	Possessive	Delete
Bi	Bislama	Bislama			Delete
Bs	Bosnian	bosanski jezik			Delete
Br	Breton	brezhoneg			Delete
Bg	Bulgarian	български език			Delete
My	Burmese	ဗမာစာ			Delete
Ca	Catalan	català, valencià			Delete
Km	Central Khmer	ខ្មែរ, ខេមរភាសា, ភាសាខ្មែរ			Delete
Ch	Chamorro	Chamoru			Delete
Ce	Chechen	нохчийн мотт			Delete
Ny	Chichewa	chiCheŵa, chinyanja			Delete
Zh	Chinese	中文 (Zhōngwén), 汉语, 漢語			Delete
Cv	Chuvash	чӳваш чӑлхи			Delete
Kw	Cornish	Kernewek			Delete
Co	Corsican	corsu, lingua corsa			Delete
Cr	Cree	ᑭᑭᑦ			Delete
Hr	Croatian	hrvatski jezik			Delete
Cs	Czech	čeština, český jazyk			Delete
Da	Danish	dansk			Delete
Dv	Divehi	ދިވެހި			Delete
Nl	Dutch	Flemish, Nederlands, Vlaams			Delete

[Update Database](#)

[Add New Language](#)

## Illustration 7: The Form for Managing Languages

There should be no need to do anything with this information as 182 languages are supported. These are all of the languages which have a 2-character ISO identifier. This table is used among other places where an apology is issued for displaying the English version of a page where no version has been defined in the user's preferred language. For example if you ask to see the main page for Canada in Spanish (lang=es) you will get:

## Canada

**There is no support for Spanish/Español.**

**Archives:**

- [Library and Archives of Canada](#)
- [Canadian Genealogy Centre \(at LAC\)](#)
- [Archives of Ontario](#)

Illustration 8: Sorry Canada does not support Spanish

## Videos

A table is added to organize information for the display of tutorial videos providing the name of the tutorial to be displayed to users depending upon their preferred language.

### Manage Tutorial Videos [? Help](#)

Pattern:

rows 0 to 9 of 9

File Name	Language	Description	Display?	Delete?
TutorialIndividual	en	a Tutorial on the Main Page for an Individual	Y	Delete
TutorialUpdateIndividual	en	a Tutorial on Editing an Individual	Y	Delete
TutorialAddChildren	en	a Tutorial on adding Children to a Family	Y	Delete
TutorialIndividual	fr	un Didactciel sur la page principale d'un Individu (piste sonoreuse en Anglais)	Y	Delete
TutorialUpdateIndividual	fr	un Didactciel sur la modification d'un individu (piste sonoreuse en Anglais)	Y	Delete
TutorialAddChildren	fr	un Didactciel sur l'addition des enfants a une famille (piste sonoreuse en Anglais)	Y	Delete
TutorialIndividual	es	un Tutorial sobre la pagina principal de una persona individual (banda sonora e	Y	Delete
TutorialUpdateIndividual	es	un Tutorial sobre la modificacion de una persona individual (banda sonora en Inç	Y	Delete
TutorialAddChildren	es	un Tutorial sobre la adicion dos infantos a una familia (banda sonora en Ingles)	Y	Delete

*Illustration 9: Manage Tutorial Videos*

## Style Classes

The original implementation defined a large number of style classes that combined styles that pertain to different aspects of the presentation. That is being replaced by combining separate simpler styles. For example:

Text color styles:

<b>black</b>	<b>black</b>
<b>same</b>	grey, used in columns to indicate values that are the same as the value of the preceding row in the table
<b>male</b>	dark blue
<b>female</b>	dark red
<b>unknown</b>	dark green
<b>error</b>	red
<b>warning</b>	dark yellow

Styles for the background color of a cell.

white	white background in input fields the user has permission to modify
odd	very bright gray almost white
even	gray, alternating with "odd" to make tables easier to read
ina	bright gray indicating the user does not have permission to modify
balloon	bright yellow PostIt Note
body	bright yellowish background used for displaying genealogical information
help	bright bluish background used for help pages

As an example where a field previously had a compound class like "actleftncerror" indicating that the data was to be displayed in such a way as to indicate that the user had permission to modify the data, was left aligned in the field, words are not capitalized, and the text is displayed in red to indicate that the value does not match the expected contents of the field, this field now has a class "error white leftnc".

## Surnames

The table of surnames now consolidates information used throughout the application, not just in the Family Tree.

**Persons with the Surname 'McLaren'** [? Help](#)

**Edit the McLaren Family Record**

IDNR:

Surname:

SOUNDEX Code:

Regular Expr. Pattern:

Notes:

Path:

Surnames are frequently spelled in many different ways either through misunderstanding by the creator of the original document or because members of the family chose to use a traditional spelling. The Surname record supports two methods for performing a search for similar surnames. One is the

SOUNDEX algorithm, originally created by American Telephone & Telegraph to support "operator assistance" calls and adopted by North American law enforcement particularly for traffic offences. This code ignores vowels and groups all consonants by broad classes except for the first letter in a name. For most names this is a useful tool. However it assumes English pronunciation and for many common names, particularly Scottish names, it matches too many surnames. The SOUNDEX code M246 matches Mackellar, Maclaren, McClarty, McClary, McClear, McCleary, McClery, McClure, McClurg, McClurtin, McElroy, McIlhargey, McIlroy, McIlwraith, McKellar, Mckeller, McLaren, McLarty, McLaurin, McLeary, and McLurg. Therefore a new mechanism has been added using the "regular expression" language to define a tighter but still loose search. The expression shown above will match "McLaren", "McClaren", "MacLaren", "MacLarren", "McLeran", and also the same names without the "Mc" prefix, which was discouraged by the English after the 1745 Jacobite rebellion.

## Given Names

A similar problem arises when searching for a match on given names. Someone christened "Elizabeth" might be known as "Lizzie", "Betty", or "Bessie". In the west of England "Peggy" and "Polly" are popular for girls christened "Margaret" and "Mary" because of the way the vocative case is formed in Welsh and Cornish. "James" is the Sassenach spelling of "Seamus", but the vocative case of that Gaelic name is "Hamish". So a table has been added to permit defining alternative searching of the family tree for matches against given names as found in original documents.

## Given Name Translation Table [? Help](#)

Pattern:

<--- rows 0 to 19 of 13 --->

Nickname	Prefix	Given Name
betreas	Bet,Beat	Beatrice
betres	Bet,Beat	Beatrice
betrice	Bet,Beat	Beatrice
betsey	Eliz,Elis,Bes,Bet	Elizabeth
betsy	Eliz,Elis,Bes,Bet	Elizabeth
bettie	Eliz,Elis,Bes,Bet	Elizabeth
bettsy	Eliz,Elis,Bes,Bet	Elizabeth
betty	Eliz,Elis,Bes,Bet	Elizabeth
elsebeth	Eliz,Elis,Bes,Bet	Betsy
elisabeth	Eliz,Elis,Bes,Bet	Betsy
elizabeth	Eliz,Elis,Bes,Bet	Betsy
hurbet	Hub,Herb	Herbert
lizabeth	Eliz,Elis,Bes,Bet	Elizabeth

## Family Tree Enhancements

The family tree portion of the site is the most developed so most of the changes had to do with exploiting the redesign of the core. Other minor issues:

- Delete citations when deleting events from from main individual record
- Blank out cause of death when deleting death
- Support abbreviations in event fields
- Correct popup of "Loading..." indicator when deleting an event that requires database action before updating the display
- Simplify addition of new events
- New events are added in chronological order so the user only has to request that events be sorted if they are out of order to begin with.
- Correct handling of updated Name event
- Suppress meaningless alerts from eventFeedback
- Ensure that when clicking "Update Person" the oldest child frame is brought to the front so it can be saved.
- Track all child frames. In particular while the "Edit Families" frame was tracked, the "Edit Parents" frame was not.



## Census Database Enhancements

Previously there was a separate script for displaying the pages used for editing, proofreading, and viewing the full contents of a census page for each of the censuses. Now there is a single script which uses separate templates for each census.

In the response to a census database query correct the sex display of name and include the requested language code in the hyperlink to Person page.

All of the pages for editing, proofreading, or displaying a full page from a census now have a button "Show Important" which hides all but the most important lines of the transcription to permit operating on a summary of the contents of the page.

## Vital Statistics Enhancements

- use class Domain to validate domain code
- use class Country to validate country code
- add fields OriginalVolume, OriginalPage, and OriginalItem to Birth and Marriage registrations to handle registrations that did not have registration numbers assigned by the Registrar.

## Bug Fixes

There are thousands of bugs fixed in the past two years and it would be pointless to list them.