February 2014 Features Update

Table of Contents

Illustration Index

Global

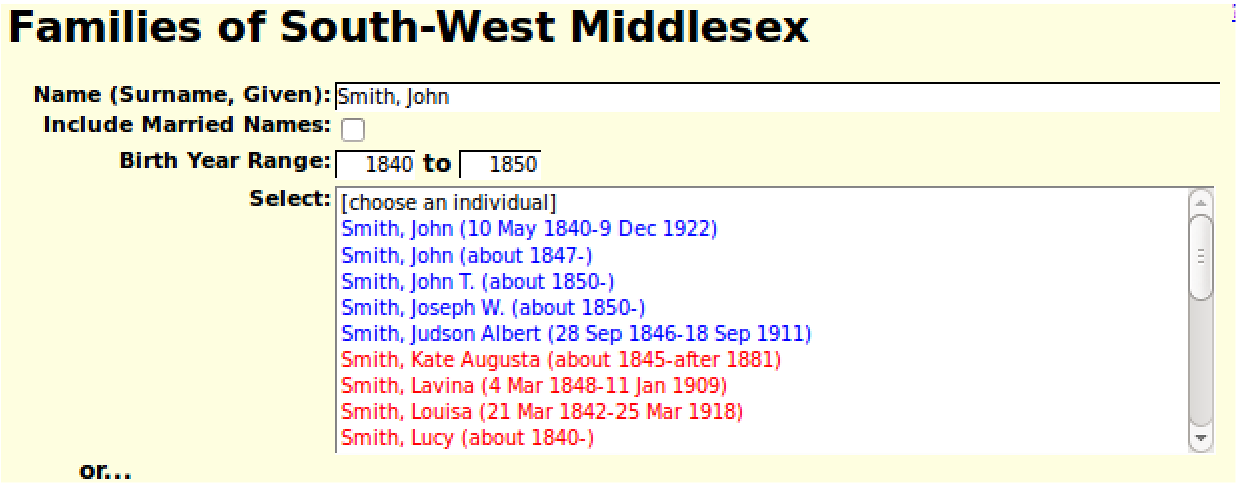
The restructuring of input forms to use CSS for layout rather than HTML tables is progressing. This also includes using the <label> tag to enclose text identifying input fields. Clicking on this label gives the input focus to the associated input field.

All scripts now support a parameter "`debug=y`". If this is passed then the script emits diagnostic information about its internal processing to assist in fixing issues.

A common routine is now used by all scripts that search the family tree for matches so that surnames are treated consistently.

Family Tree Enhancements

The main search by name page is enhanced to provide the ability to restrict the responses by birth date. This is particularly useful with extremely common names:



Families of South-West Middlesex

Name (Surname, Given): Smith, John
Include Married Names: □
Birth Year Range: 1840 to 1850
Select: [choose an individual]
Smith, John (10 May 1840-9 Dec 1922)
Smith, John (about 1847-)
Smith, John T. (about 1850-)
Smith, Joseph W. (about 1850-)
Smith, Judson Albert (28 Sep 1846-18 Sep 1911)
Smith, Kate Augusta (about 1845-after 1881)
Smith, Lavina (4 Mar 1848-11 Jan 1909)
Smith, Louisa (21 Mar 1842-25 Mar 1918)
Smith, Lucy (about 1840-)
or...

Illustration 1: Nominal Index with Birth Year Range

Several scripts return information about an individual in the form of an XML document. This permits the information to be processed by application code, either as part of this system or potentially by a smart-phone applet. This information has been enhanced to include all event information. Each event appears as an <event> tag. For example a birth event might appear as:

```
<event ider="2705">
 <idir>2705</idir>
 <ider>2705</ider>
 <idtype>Individual</idtype>
 <idet>2000</idet>
 <eventtype>born</eventtype>
 <eventdc>500000188100001901</eventdc>
 <eventd>between 1881 and 1901</eventd>
 <eventsd>18810000</eventsd>
 <idlrevent>1</idlrevent>
 <eventloc/>
 <idarevent>0</idarevent>
 <desc/><description/>
</event>
```

The event information included in the response includes any source citations for the event. For example:

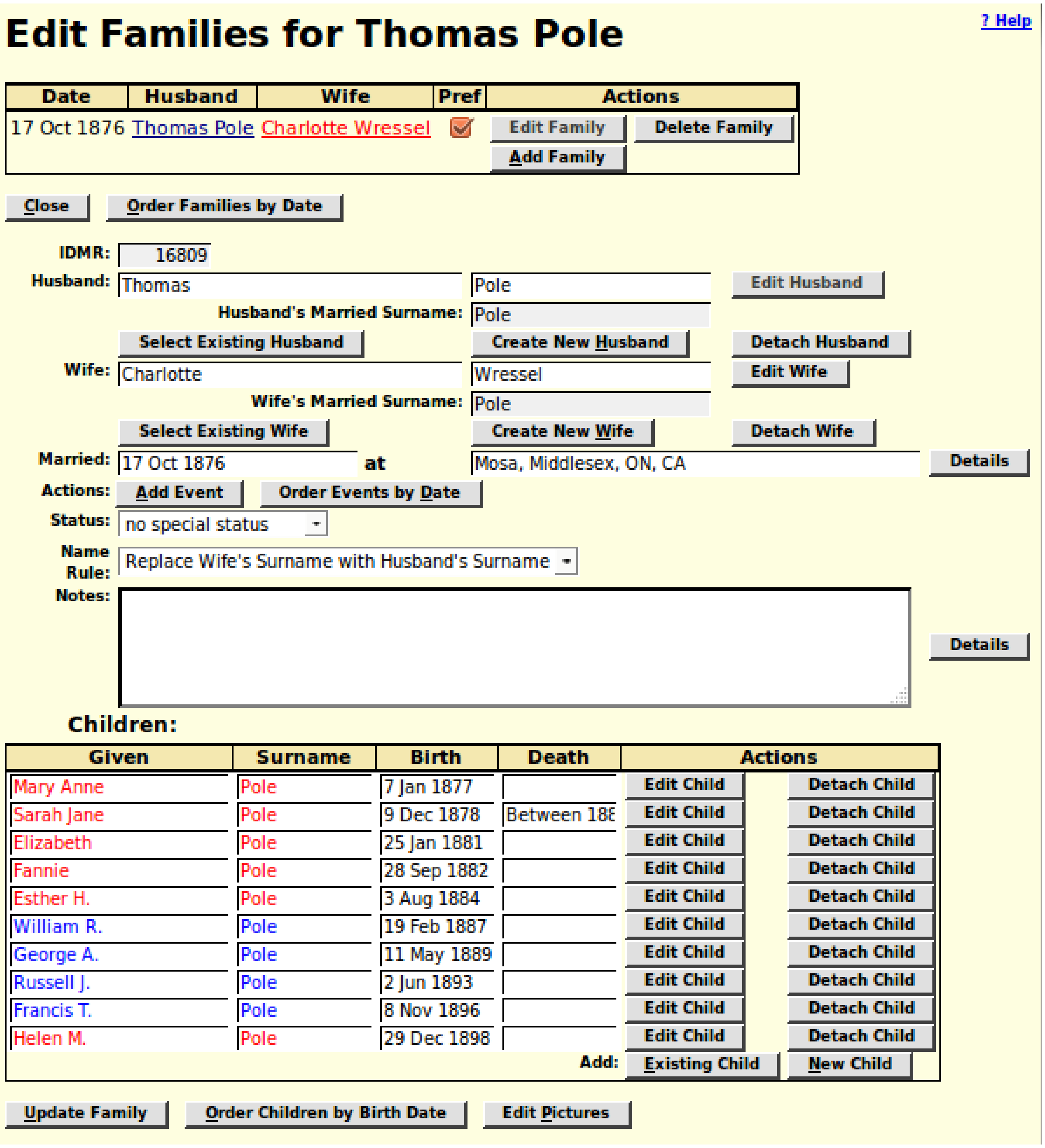
```
<event ider="1367">
 <ider>1367</ider>
 <idir>3705</idir>
 <idet>47</idet>
```

```
  <eventtype>worked as a </eventtype>
  <order>0</order>
  <eventd>1861</eventd>
  <idlrevent>325</idlrevent>
  <eventloc>Delaware, Middlesex, ON, CA</eventloc>
  <desc/>
  <gedtag/>
  <eventexclude>N</eventexclude>
  <idtype>Individual</idtype>
  <idar>0</idar>
  <description>Baker</description>
  <sentenceoverride/>
  <qstag>N</qstag>
  <rgexclude>N</rgexclude>
  <citations>
    <citation idsx="5696">
      <idsx>5696</idsx>
      <idsr>12</idsr>
      <idime>1367</idime>
      <type>30</type>
      <srcdetail>CW, dist 22 Middlesex, subdist 239 Delaware, div 1 page 7</srcdetail>
      <srcsurety>3</srcsurety>
      <srcprint>1</srcprint>
      <srcprintdetail>1</srcprintdetail>
      <srcprinttext>0</srcprinttext>
      <srcdettext/>
      <srcdetnote/>
      <srcprintnote>0</srcprintnote>
      <enteredd/>
      <enteredsd>-99999999</enteredsd>
      <filingref/>
      <order>0</order>
      <used>0</used>
      <verified>0</verified>
      <content/>
      <override/>
      <overridefootnote>0</overridefootnote>
      <overridesubsequent>0</overridesubsequent>
      <overridebibliography>0</overridebibliography>
    </citation>
  </citations>
</event>
```

To see the complete XML output for an individual go to any individual in the family tree and replace “legacyIndivid” in the address line at the top of the browser window with “getIndivXml” and add the additional parameters “&includeFamilies=Y&includeParents=Y”. As a result of this enhancement an XML style-sheet could be implemented that would come close to duplicating the output of the main script legacyIndivid.php.

The initial setting of the 'order' field for a citation is moved into the constructor for the class LegacyCitation so setting this field is performed consistently.

The functionality of the scripts editMarriages.php and editParents.php has been very significantly enhanced to simplify managing the family and fix significant missing functionality. The most common changes to the members of the family, changing names and birth and death dates, can now be performed entirely in these forms.



? Help

Edit Families for Thomas Pole

Date	Husband	Wife	Pref	Actions	
17 Oct 1876	Thomas Pole	Charlotte Wressel	☑	Edit Family	Delete Family
				Add Family	

Close   Order Families by Date

IDMR:	16809		
Husband:	Thomas	Pole	Edit Husband
	Husband's Married Surname:	Pole	
	Select Existing Husband	Create New Husband	Detach Husband
Wife:	Charlotte	Wressel	Edit Wife
	Wife's Married Surname:	Pole	
	Select Existing Wife	Create New Wife	Detach Wife
Married:	17 Oct 1876 at	Mosa, Middlesex, ON, CA	Details
Actions:	Add Event	Order Events by Date	
Status:	no special status		
Name Rule:	Replace Wife's Surname with Husband's Surname		
Notes:			Details

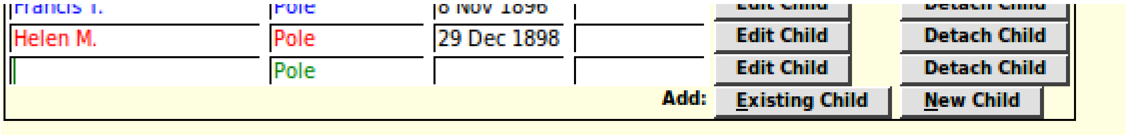
Children:

Given	Surname	Birth	Death	Actions	
Mary Anne	Pole	7 Jan 1877		Edit Child	Detach Child
Sarah Jane	Pole	9 Dec 1878	Between 188	Edit Child	Detach Child
Elizabeth	Pole	25 Jan 1881		Edit Child	Detach Child
Fannie	Pole	28 Sep 1882		Edit Child	Detach Child
Esther H.	Pole	3 Aug 1884		Edit Child	Detach Child
William R.	Pole	19 Feb 1887		Edit Child	Detach Child
George A.	Pole	11 May 1889		Edit Child	Detach Child
Russell J.	Pole	2 Jun 1893		Edit Child	Detach Child
Francis T.	Pole	8 Nov 1896		Edit Child	Detach Child
Helen M.	Pole	29 Dec 1898		Edit Child	Detach Child
			Add:	Existing Child	New Child

Update Family   Order Children by Birth Date   Edit Pictures

Illustration 2: Edit Families Form

In particular the “New Child” button formerly popped up a window to display the Edit Individual script for a new individual. That script is now displayed only if you click on the “Edit Child” button. The “New Child” button now just adds a line to the end of the list of children as shown in the following illustration.



Francis T.	Pole	8 Nov 1896		Edit Child	Detach Child
Helen M.	Pole	29 Dec 1898		Edit Child	Detach Child
	Pole			Edit Child	Detach Child
			Add:	Existing Child	New Child

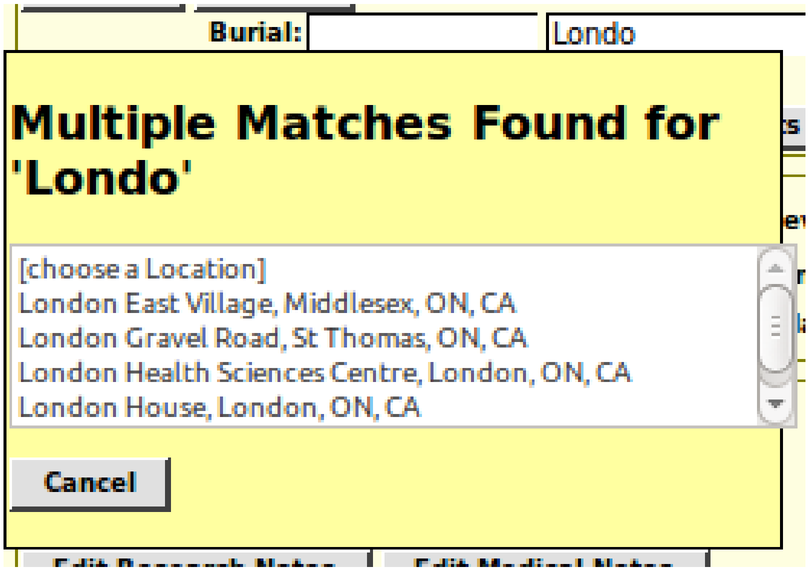
Illustration 3: Adding a Child to the Marriage Form

You can enter the name and birth and death dates of the child directly into that new line, and then continue to add more children. When you finally click on the “Update Family” button all of the changes you have entered into this form are applied. Note that there is no explicit way to set the gender of the individual on this form. However if you enter a given name that contains one of the most common given names of the time the gender is set. This is shown by changing the colour of the text from green to either red or blue. If you wish to postpone invoking the detailed edit form to explicitly set the gender you can just temporarily add one of the common English given names, such as”John” or “Mary” to set the gender, and then remove it.

The “Order Children by Birth Date” button now works in all cases. There was a design deficiency that you could not sort the list if you had added an existing child to the list. You had to add the existing child, update the family, then edit the family again to order the children.

Note that if you accidentally add too many new children, and some of the lines are still blank, and green, except for the default surname, the update ignores them.

The support for completing location names in all of the dialogs which support them has been enhanced to be easier to follow. Note that because you are accessing these forms through your browser, and you have probably accepted the default browser behavior to prompt you with values from your browser history when entering data into a field, you get two assists. The browser feature follows your keystrokes and if you pause for a second or two it displays a list of past values that you have entered into that field that most closely match the text you have entered so far. You can select one of those values with your mouse. However if you move the focus out of an input field that contains a location, for example by tabbing to the next field, then the application takes over and uses the text you have entered to perform a lookup of the locations table of the database. This is designed to ensure that the number of different versions of the same actual location is minimized. For example if you just type “Metcalfe” the field is expanded automatically to “Metcalfe, Middlesex, ON, CA”, which is the standard form for specifying that township in the database. If no existing locations match a dialog pops up to warn you that your will be creating a brand new location. You can then check to make sure there are no spelling mistakes before accepting the new location. If more than one existing location entry matches the text you have entered then you are given a choice. That choice used to be presented by inserting a selection list directly into the form, however it is now presented in a popup dialog as shown in Multiple Matches for a Partial Location Name on page 8:



Multiple Matches Found for 'Londo'

[choose a Location]
London East Village, Middlesex, ON, CA
London Gravel Road, St Thomas, ON, CA
London Health Sciences Centre, London, ON, CA
London House, London, ON, CA

Illustration 4: Multiple Matches for a Partial Location Name

Census Database Enhancements

All of the census related pages and scripts are now in the folder “database” including the table of contents.

Vital Statistics Enhancements

The internal interface between the Vital Statistics pages and the database has been changed to use object oriented classes `Birth`, `Death`, `Marriage`, and `MarriageParticipant`. These classes exploit the support previously implemented for the family tree and census records. This change means, among other enhancements, that when a BMD record is updated only fields whose value has changed are written to the database, and that updates to the vital statistics tables are logged for recovery.

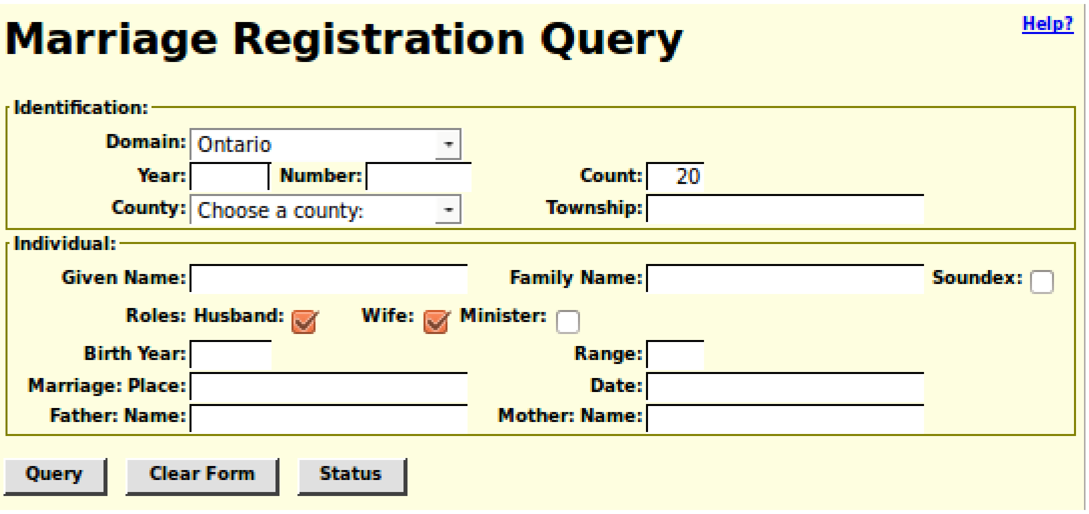
Although there are no records for any other administration beside Ontario in the current database support has now been added for recording vital statistics records from all of the Canadian provinces plus Michigan state in the USA.

The scripts `BirthRegUpdate.php`, `MarriageRegUpdate.php`, and `DeathRegUpdate.php` have been enhanced to use the class `LegacyCitation` to add citations into the family tree rather than writing them directly to the database. This ensures consistent handling, and that the changes are logged.

In the list of potential matches to the deceased individual when editing a death registration the maiden name is moved before the married name to be consistent with everyday usage.

In the display of all vital statistics, where a matching individual in the family tree has been identified for a participant, the hyper-link to the individual record now bears the full name, birth date, and death date from the database, rather than just echoing the input fields from the form.

The form for searching the database of marriage registration transcriptions has been redesigned to provide additional functionality and clearer structure:



Marriage Registration Query

Help?

Identification:

Domain: Ontario

Year: Number: Count: 20

County: Choose a county: Township:

Individual:

Given Name: Family Name: Soundex: □

Roles: Husband: ☑ Wife: ☑ Minister: □

Birth Year: Range:

Marriage: Place: Date:

Father: Name: Mother: Name:

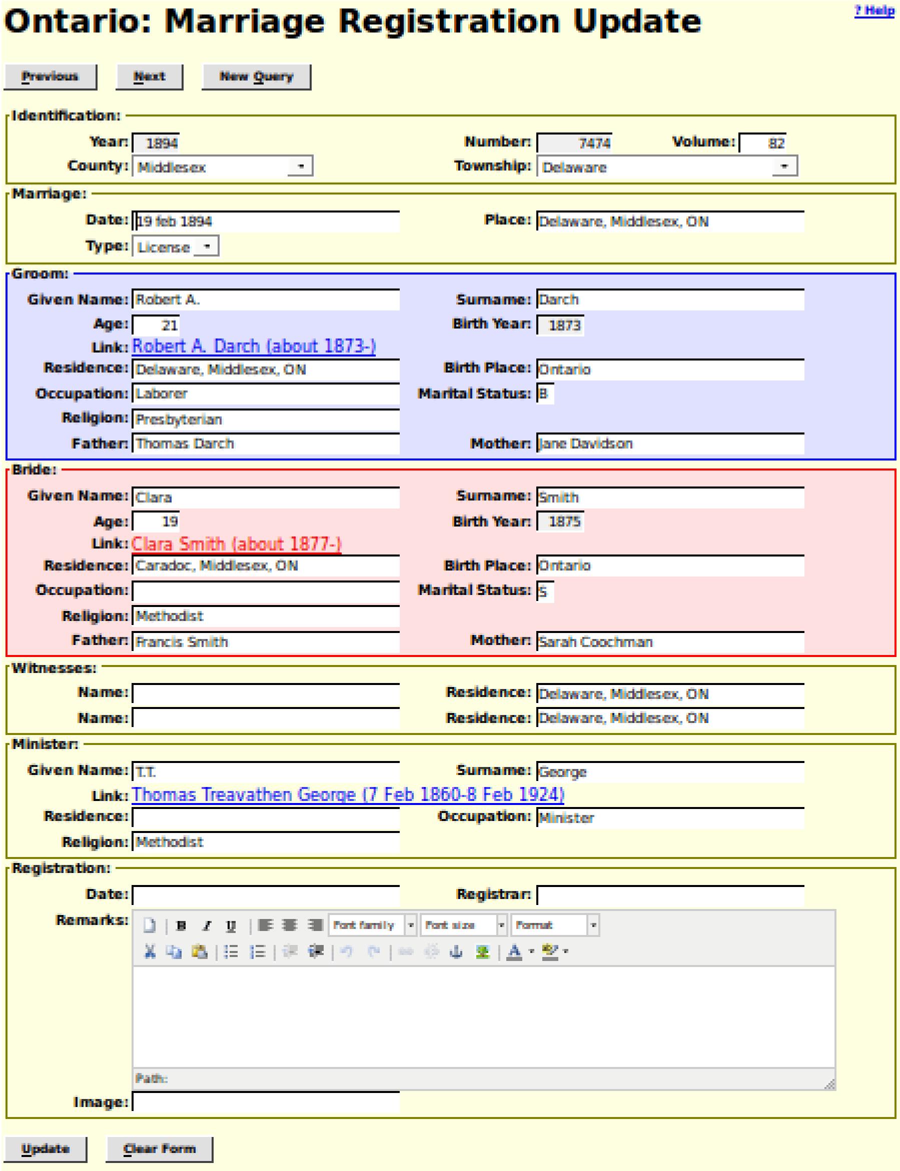
Illustration 5: Marriage Registration Query

The identification portion now supports administrative domains other than Ontario, although no records have been transcribed for other domains. The input fields are grouped by their role in the query.

The appearance of the page for updating a marriage registration has been substantially redesigned as part of the implementation of support for HTML 5. Each of the major sections of this quite complex form is enclosed in an identifying frame. More information can now be recorded about the minister.

The list of potentially matching individuals for a participant in the marriage now includes the names of their parents, if known, which can be compared to the names of the parents that appear in the record. The support for finding matching individuals in the family tree has been extended to include the presiding minister. Only individuals who have an occupation event indicating they were a minister, priest, or clergyman are included in the results.

The form now looks like Illustration 6: Edit Marriage Registration Dialog on page 11:



Ontario: Marriage Registration Update

Identification:

Year: 1894 Number: 7474 Volume: 82
County: Middlesex Township: Delaware

Marriage:

Date: 19 feb 1894 Place: Delaware, Middlesex, ON
Type: License

Groom:

Given Name: Robert A. Surname: Darch
Age: 21 Birth Year: 1873
Link: Robert A. Darch (about 1873-)
Residence: Delaware, Middlesex, ON Birth Place: Ontario
Occupation: Laborer Marital Status: B
Religion: Presbyterian
Father: Thomas Darch Mother: Jane Davidson

Bride:

Given Name: Clara Surname: Smith
Age: 19 Birth Year: 1875
Link: Clara Smith (about 1877-)
Residence: Caradoc, Middlesex, ON Birth Place: Ontario
Occupation: Marital Status: S
Religion: Methodist
Father: Francis Smith Mother: Sarah Coochman

Witnesses:

Name: Residence: Delaware, Middlesex, ON
Name: Residence: Delaware, Middlesex, ON

Minister:

Given Name: T.T. Surname: George
Link: Thomas Treavathen George (7 Feb 1860-8 Feb 1924)
Residence: Occupation: Minister
Religion: Methodist

Registration:

Date: Registrar:
Remarks:
Image:

Illustration 6: Edit Marriage Registration Dialog

Bug Fixes

- The script legacyIndivid.php, which implements the main page for an individual, no longer uses the obsolete method LegacyIndiv::getChildRecord.
- The script LegacySource.php, which implements the detailed display of the contents of a source record, is changed to improve parameter validation, in particular to not fail if invoked with an invalid IDSR key value, and to display errors detected as a result of parameter validation as a highlighted error message.
- The script EditSource.php, which implements the ability to alter the contents of a source record, is changed to improve parameter validation, in particular to not fail if invoked with an invalid IDSR key value, to display errors detected as a result of parameter validation as a highlighted error message, use CSS to layout the form, and improve the appearance of selection lists.
- The script updateSource.php, which applies the changes requested by the script EditSource.php, is enhanced to use the new ability of the LegacySource class to create a source by name.
- The script QuerySurnames.php now accepts its named parameters with any case.
- The form for editing a page of the 1871 census was missing default values for several columns because of text accidentally copied from the 1861 census form script.
- One of the URLs referenced in the hierarchy of links in the top and bottom of the script CensusUpdateStatusDist.php, which displays a summary of all of the enumeration divisions in a census district, was incorrect.
- In the script MarriageRegStats.php, which presents a year by year summary of the transcription effort on marriage registrations, the “Help?” link was positioned incorrectly.
- The script MarriageRegDetail.php did not display information on the presiding minister if the user was not signed in as a contributor.
- There were a couple of minor HTML syntax errors in the output of the script DeathRegDetail.php. Most browsers handled the errors in a way that hid the problem from users.
- In the script BirthRegDetail.php, which displays the details of the transcription of a birth registration, there was an HTML error in the presentation of the sex indicator which meant that the sex from the database was not displayed in the edit form.
- The scripts BirthRegResponse.php, DeathRegResponse.php, and MarriageRegResponse.php, which display the results of a search for matching vital statistics registrations, produced invalid HTML if the SQL command failed. Also there were missing items from the hierarchy of links in the top and bottom of the page.
- There was a bug in the script editPicture.js that caused the dynamic functionality of the page to not be initialized.

- When the script editPicture.php is invoked the dimensions of the window are now set so the entire form is visible without scrolling.