

User Guide

SelCtl Land Trust Database

Larry Bednar

July 2, 2004

Purpose

This document provides directions for use of the database developed for a land trust non-profit during late 2003/early 2004.

Description

The database is constructed using Microsoft Access 2000. As much as possible, automated functions provided by the application are encapsulated in VisualBasic programming associated with forms and reports, or in VisualBasic code modules included with the application.

The application consists of two major components: a “server” database that functions to store the data, and a “client” database that provides forms, reports, and automation function useful to the general user. Although the general user will work directly with the “client” application, all data is actually stored in the “server”. It is expected that the “server” will be located on a computer accessible over a local area network. All individuals working with the application will access a single copy of the “server” database, ensuring that everyone has access to the same data.

The database is designed to facilitate tracking of information in the following data subjects:

Parties – A person, family or organization about which detailed information such as address, phone number, involvement in easement projects, and donations will be tracked.

Party Categories – Classifications assigned to parties. An arbitrary number of categories may be assigned to each party.

Party Group Memberships – Membership of parties in groups defined for use by the organization, with start and end dates assigned. (“newsletter”, “annual report”, etc.). An arbitrary number of group memberships may be assigned to each party.

Party Locations – Location information, including address and location type designation. Additionally, each location may be assigned a “county” or “region” code. An arbitrary number of locations may be stored for each party.

Persons – An individual who is associated with one or more “parties”. A “person” does not have any information other than relationships with parties and other persons tracked in the database.

Party Members – Links between persons and parties they are associated with. A single person may be associated with any number of parties, with a designated “role” in each party. A single “party” may have an arbitrary number of persons associated with it.

Related persons – Relationships between one person and another, and the type of relationship (“sibling”, “father”, “son”, “spouse”, “friend”, etc.)

Easement Projects – Summary information about an easement project, significant milestones achieved during the project, staff roles, and involved parties.

Donations – Information about standard donations, “proxy” donations, and expected future donations.

Pledges – Promised of future donations

Contacts – Interactions between staff members and parties, such as email, phone conversation, etc. Each contact may be assigned an arbitrary number of codes to indicate the subject of the interaction. Each party and staff member may have an arbitrary number of contacts associated.

A number of reports that seemed to be of obvious utility have been included in the first delivery of the application. A number of additional reports may be identified as desirable after users familiarize themselves with the database.

The application was designed from the start to be operated in a client/server fashion. A number of high-level design features therefore differ from approaches that might be appropriate and common in development of single-user MS-Access databases.

In addition, a number of fairly advanced VisualBasic tools have been implemented within the application in order to streamline later modification/revision of the application.

Implementation of additional forms or reports that use the same approaches already used in the application will require intermediate to advanced SQL programming skills and intermediate level familiarity with VisualBasic and the event model applied to forms and reports in MS-Access.

Repair of problems with VisualBasic code provided with the module should probably only be undertaken by an advanced VisualBasic programmer. The interactions between functions and procedures in several of these tools are complex, and many of these tools are applied frequently and widely during use of the application. It would be easy for an unskilled VisualBasic programmer to inadvertently make a single change that would result in major difficulties in many areas of the application.

Installation

The computer used to run the application must have a copy of MS-Access 2000 or later installed. The MS-Access “Linked Table Manager” add-in should be installed, as this will be needed. This computer must also be capable of access the copy of the “server” database being shared by database users.

Typically, a user will place a copy of the “client” application on their computer. The copy provided will be obtained from the administrator and will automatically provide the access level appropriate for the user.

The first time the “client” application is started, the user may receive error messages that result from the “client” database being unable to find the “server” database. (Ideally, the administrator will be involved in this initial start-up.) This typically occurs because the application must be provided with the location of the “server” application where the data is held – a “location” value entered on another computer often will not be valid from the new computer.

If error screens are displayed during first use, the user should:

1. Use “ok” or “end” responses to the displayed error screens (a series of 2-3 responses of this type might be required)
2. Once the “main switchboard” form is displayed without overlaid error message windows, use the MS-Access “linked table manager” add-in (accessible from MS-Access drop-down menus “Tools/Database Utilities...”) to indicate the location of the “server” database.

3. After specifying the location of the “server” application, shut down the client application and reopen it. The user should be immediately presented with a “main switchboard” form, with no windows displaying error message, etc.

Functions for the General User

This section outlined major functions provided for the general database user.

“Main Switchboard” Form

The “main switchboard” form is the “hub” of use for the general database worker. All functions typically needed by the user should be accessible from this form. There may be unusual occasions when the user will need to use a different approach to accomplish some task, but it is recommended that the main switchboard be used as much as possible.

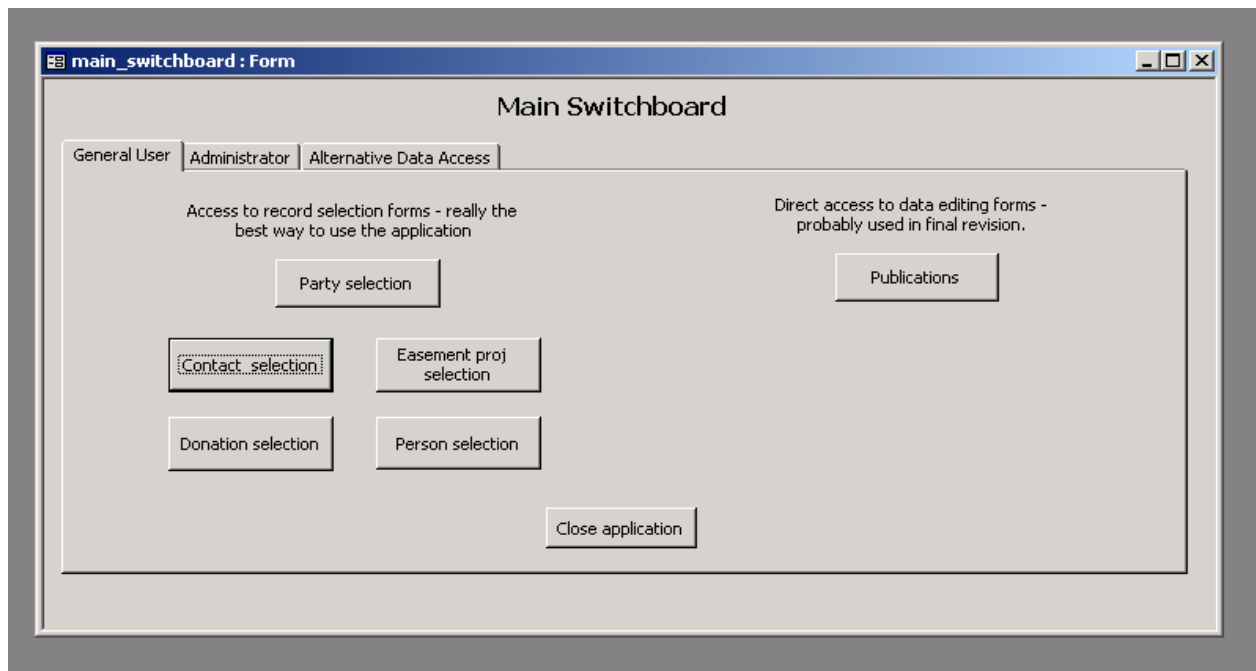


Fig 1: “Main Switchboard” Form

Most automation designed to facilitate the user’s work with the database is coordinated with actions provided on the main switchboard. Alternative approaches may not engage automation features as reliably.

“Record Selection” Forms

“Record Selection” forms are provided for each major “data subject”. Each of these forms is designed to allow a fast search for existing records that match a number of search criteria specified by the user.

The screenshot shows a software window titled "party_selection : Form". Inside, the main heading is "Party Selection". At the top left, there is a label "Selected rows:" followed by a text box containing the number "5608". To the right of this is an "Exit" button. Below the "Selected rows:" label, there are three search and selection fields: "Party name search pattern:" with an empty text box, "Party type:" with a list box containing "business", "family", "foundation", and "individual", and "Party category:" with a list box containing "(none)", "ag group", "annual rpt", "appraiser", "art auction", "art auction attendee 9", and "art catalog". To the right of these fields, there is a "Party member" section with a "Person name search pattern:" text box. Below that is a "Group membership" section with a "Group:" list box containing "annual report", "art show artist 2003", "art show sponsor 2003", "Bills fishing list", "Devils kitchen group", "Jays fishing list", and "newsletter". Below the "Group:" list box is an "Effective date:" text box and an "Apply" button. At the bottom right of the "Group membership" section is a "Location" section with a "Location region:" list box containing "beaverhead v", "big hole v", "big horn v", "bitterroot v", "blackfoot v", "boulder river v", and "bull mtns". At the bottom left, there is a "Selection filter:" label followed by a large empty text box. At the bottom right, there are "Display" and "New" buttons.

Fig 2: Example “Record Selection” Form

“Record Selection Display” Forms

Each “record selection” form is associated with a coordinated form that displays a concise representation of the records matching the criteria specified by the user. Typically, this form is reached by clicking the “display” button on a record selection form.

party_selection_display : Form

Party - Selected Records

Selected rows: 9

Selection description:
(associated rows in pc with 'party_category_defn_id' in at least one of (bob marshall1))

Selected records

party_id	name	add_date	party_type	staff_mbr
3866	Diana Neely	2/23/2004	individual	none
3918	Carolyn Dietrich	2/23/2004	individual	none
3960	Sherry O'Hearn	2/23/2004	individual	none
3970	Cas Still	2/23/2004	individual	none
3978	Jill Watkins	2/23/2004	individual	none
4095	Lisa Mitts	2/23/2004	individual	none
4103	Susan K Dietrich	2/23/2004	individual	none
4129	Mary S. Nance	2/23/2004	individual	none
109	Amy Eaton Royer	2/23/2004	individual	none

Revise selection New Edit selected Report choice: Report Exit

Fig 3: Example “Record Selection Display” Form

Editing An Existing Record

In cases where the user wants to edit a specific record, the user simply selects the desired record from among those displayed on this form. The “edit selected” button will open the appropriate editing form with the record selected by the user ready for modification.

Adding a New Record

Alternatively, the user may review the displayed records and decide that none of them represents the data to be entered. A new record would then be appropriate. The “new” button will open the appropriate editing form on a blank record ready for entry of data by the user.

Generating a Report

Another alternative is that the user may want to represent the entire selected set of records using a pre-defined report. In this case, the user may select the report to be used using the drop-down list provided near the bottom of the form. Only those reports appropriate to the subject area will be listed. The “report” button will result in the generation of a report using only the records displayed on the form.

Reports

Reports are typically accessed only through the “record selection”/“record selection display”/“report” process. Direct access to reports would typically only be used by developers or administrators.

Most reports are designed for printing or screen display. However, some are provided to assist in export of data to spreadsheet formats, etc. These reports typically display a representation of the data in a standard report format. This can be reviewed by the user to ensure that the proper set of records is represented. A pop-up window that appears at the same time as the report allows the user to either cancel the report or continue to export the represented data to an Excel spreadsheet, etc.

Functions for the Administrator

Several functions provided in the application are sensitive enough to be reserved for the database administrator. These tasks are generally those where a technical understanding not expected of the general user is either mandatory or at least helpful:

Distribution of “Client” Application Copies with appropriate access settings

The administrator should accept full responsibility for setting the access level assigned to each installed copy of the client application.

At present, the application uses a very simple security mechanism through which each installed copy of the client application is set to one of three access levels:

1. Administrator – Has access to all forms in the application
2. Financial – No access to “administrator” functions such as ability to edit code definitions.
3. Gen user – No access to “administrator” functions such as ability to edit code definitions. Access to donations/pledges is “read-only”.

The access level is set by VisualBasic statements in the InitApp module, and is displayed in the upper right corner of the “Main Switchboard” form.

Any copy of the client application may be set to any of the three access settings by a competent VisualBasic programmer.

Alternatively, the administrator may choose to keep three “master” copies of the client application – one set at each security level. Each user may then be provided with a copy of the version providing appropriate access.

It is possible to set a more standard (and more rigorous) security scheme using Microsoft workgroups. This requires more work on the part of the administrator, but offers the opportunity to assign each individual user a unique account name and password. Access to various parts of the application may then be controlled based on the identity of the user. No effort was made to implement a more complex security approach such as this until a clear need was demonstrated, and staff are confirmed to have appropriate administration skills.

Installation of Copies of “Client” Application

The administrator should assist users in initial installation of the “client” application, verifying correct operation, specifying the location of the “server” database, etc. as needed. This is a simple operation for someone who has been through it before, but can be confusing for less familiar users.

Code Definition

It is recommended that only the database administrator perform code administration.

The following principles should be applied:

1. Use abbreviations consistently.

Ideally, every time a word or phrase is abbreviated, it should use the same abbreviation.

2. Use a consistent approach to constructing abbreviations

If all existing “art auction” codes *start* with “art auction...”, newly added codes should probably use the same pattern. These consistencies in pattern make it easier for the user to apply codes.

3. Be cautious about modifying existing codes in a way that changes their basic meaning

Changes in spelling, use of abbreviations, etc. can be readily made if the basic meaning of the code remains the same. Once these changes are made, they will show up everywhere in the application where the code has been used.

However, a change that alters the basic meaning of an existing code may cause problems since any existing uses of that code would also be changed. This may alter the meaning of data already entered by users in a way that is inaccurate.

All uses of codes must be removed from the database before a code may be deleted. Stating this a different way, no code may be deleted that is still in use anywhere in the application.

NOTE: Changes in existing definitions for “donation type” are of particular sensitivity. In several places in the application, form functions such as enabling/disabling of data entry are dependent on which code has been entered by the user. If the code “funds” is changed to some other value, the resulting form behavior may not be appropriate for the revised codes.

4. Perform code changes at a time when users are not using the database

The “client” application loads copies of codes into local versions of the code definition tables at start-up. This action is taken to all the “client” application to access these code values locally, rather than querying the “server” for that information each time it is required. This step should enhance performance by reducing the amount of data passed over the local area network.

However, this also means that the user typically will not have a current copy of the codes if you make a change after they have already started the “client” application. Users in this situation will need to close and restart the “client” application to ensure that they have a current copy of the code set.

If the administrator only makes code changes when no users are working, this problem is avoided.

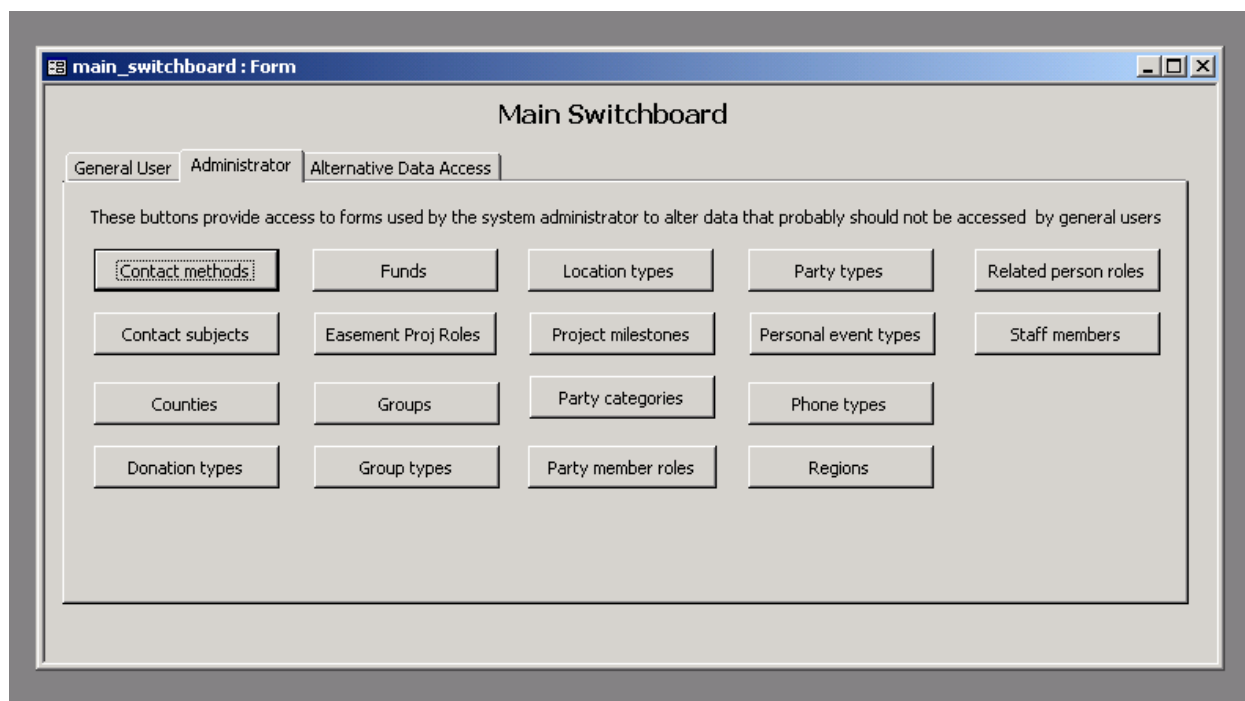


Fig 4: Access to “code editing” functions by main switchboard

“Rpt_row” Table Purging

The administrator should occasionally check the contents of the rpt_row table on the “server” application. Ideally, this should be done when no users are working with data. Typically, this table should be empty. If rows are present and no users are connected, the following steps should be taken:

1. Make note of the fact that rows are present and contact the developer. This may indicate a minor problem that requires attention.
2. Delete all rows in the table.

Alternative Access to Editing Forms

In the first version of the application provided, the ability to directly access the main data editing forms is provided.

However, it is recommended that direct access to these forms should not be a typical use approach for general users, as this approach results in passing of greater amounts of data across the local area network from server to client. This may result in degraded performance when several users are using the database or performing other work that passes data through the network.

The suggested approach to general use is instead to always access data using the “record selection” forms provided with the application. This approach will result in reduced introduction of duplicate records into the database, and will typically be faster for the user while simultaneously reducing the amount of data passed through the network.

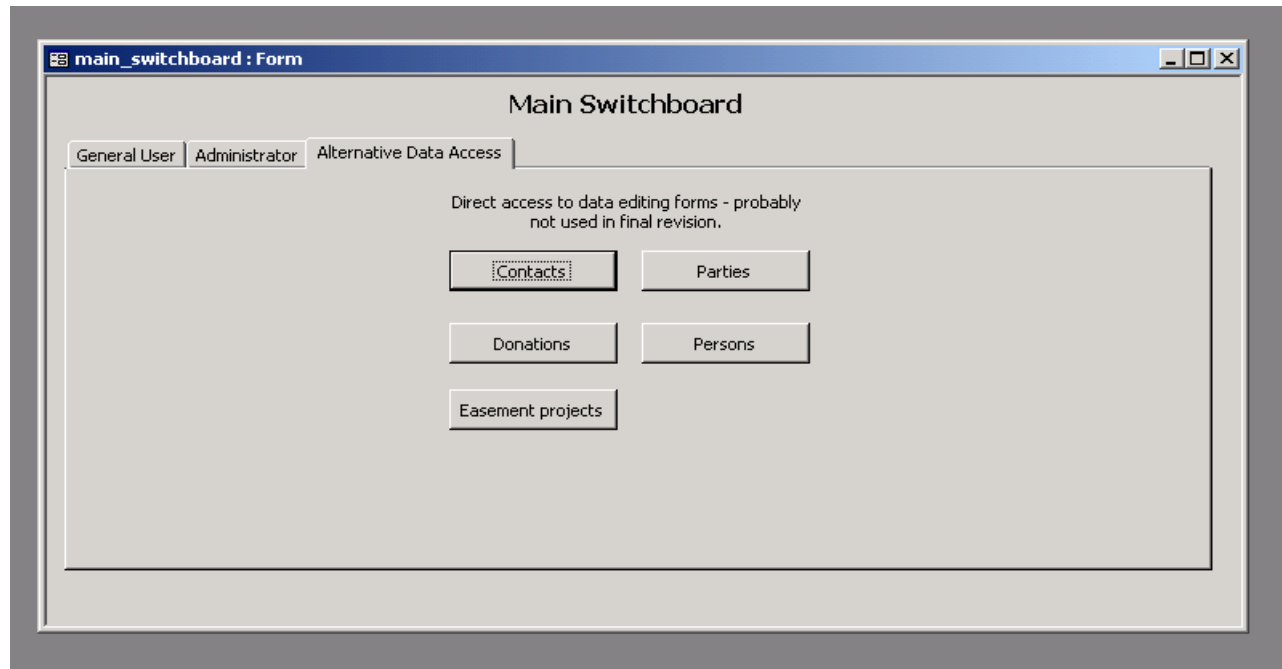


Fig 5: Access to “alterative editing” functions by main switchboard

Report Descriptions

This section contains a brief description of reports that are included with the system. Reports are accessed from the “report choice” drop down list provided at the bottom of each “record selection display” form. Once the user selects a report using this list, the button labeled “report” is enabled. Clicking the “report” button then will display the report, with all records shown in the “record selection display” form represented.

Contact Reports

These reports describe “contacts” entered into the system.

Contact_rpt

This report is designed to display a chronological list of all contacts with parties represented in the selected group of contacts.

The displayed contact records are ordered by party, and then in chronological order. .

Donation Reports

These reports describe “donations” entered into the system.

Donation_rpt

This report provides a short description of each selected donation.

The displayed contact records are ordered by “final donor party name”, and then in chronological order. .

Donation_summary_rpt

This report summarizes all selected donations by party and fund.

The grand total for each party (across all funds) and for all selected donations is also provided.

Easement Project Reports

These reports describe “easement projects” entered into the system.

Emnt_proj_land_steward_party_rpt

This report is designed to display a list of all parties associated with the selected group of easement projects. Address information for the related parties is also displayed.

Emnt_proj_land_steward_rpt

This report is designed to display the easement projects and associated parties for all land stewards associated with the displayed set of easement projects.

Emnt_proj_landowner_rpt

This report is designed to display a list of easement projects associated with each land steward associated with the displayed set of easement projects.

The displayed list of projects for each land steward is ordered by region.

Party Reports

These reports describe “parties” entered into the system.

Party_address_export_rpt

This report is designed to export addresses for all locations related to the selected group of parties. The first and last name of the primary person associated with the party is also exported.

The resulting file should be easily used by word processing software such as Microsoft Word in letter merges or printing of mailing labels.

By default, the exported file will be a Microsoft Excel file. The created Excel workbook will include field names in the first row. The user may specify an alternative file name and location, or an alternative file format, as desired.

Party_group_mbr_rpt

This report lists the associations of the selected parties with groups, as represented by “group membership period” records. The report is ordered by group name, with parties belonging to each group listed within each group.

Party_loctn_rpt

This report lists information (primarily addresses) for all locations associated with the selected group of parties.

Party_primary_address_export_rpt

This report is designed to export addresses for the “primary” location associated with each of the selected parties. Information about locations that are not designated as “primary” locations is excluded.

The resulting file should be easily used by word processing software such as Microsoft Word in letter merges or printing of mailing labels.

By default, the exported file will be a Microsoft Excel file. The created Excel workbook will include field names in the first row. The user may specify an alternative file name and location, or an alternative file format, as desired.

Party_primary_loctn_labels_avery5162

This report is designed to create mailing labels for the selected group of parties.

The created labels use the “party name” as the first line of the label, and use the address of the designated “primary” location. Labels are sorted by zip code value.

The report is formatted to match Avery 5162 mailing labels (14 rows per sheet in two columns). .

Party_primary_loctn_labels_avery5660

This report is designed to create mailing labels for the selected group of parties.

The created labels use the “party name” as the first line of the label, and use the address of the designated “primary” location. Labels are sorted by zip code value.

The report is formatted to match Avery 5660 mailing labels (30 labels per sheet in 3 columns).

Party_primary_person_labels_avery5162

This report is designed to create mailing labels for the selected group of parties.

The created labels use the name of the person designated as the “primary” party member as the first line of the label, and use the address of the designated “primary” location. Labels are sorted by zip code value. If there is no designated “primary member” for a party, that row is excluded from the report.

The report is formatted to match Avery 5162 mailing labels (14 rows per sheet in two columns). .

Party_primary_person_labels_avery5660

This report is designed to create mailing labels for the selected group of parties.

The created labels use the name of the person designated as the “primary” party member as the first line of the label, and use the address of the designated “primary” location. Labels are sorted by zip code value. If there is no designated “primary member” for a party, that row is excluded from the report.

The report is formatted to match Avery 5660 mailing labels (30 labels per sheet in 3 columns). .

Person Reports

These reports provide an organized presentation of information concerning a selected group of persons displayed on the “person selection display” form.

Person_party_mbr

This report lists all parties that each selected person is a member of.

Developer Support

A developer working on maintenance or alteration of this application should find supporting materials have been provided.

The following documents should be helpful to an individual working on the application itself:

1. Entity-relationship diagrams.

Diagrams representing the data structures of the database. These are provided in PDF format although the actual diagrams were developed and stored in a software data modeling tool that allows automatic generation of table creation scripts.

2. Data model reports

RTF documents providing a detailed outline of database objects used in the application.

3. Form_module_modification document

Step by step guidelines to modifying/copying form or report modules in a fashion that integrates fully with the tools provided in the application.

4. Developer guide

Provides a high-level explanation of the functions performed by VisualBasic code modules and form/report modules, outline of naming conventions, some explanation of design choices made during development. Also provides some explanation of internal mechanisms employed by application components.

5. UML sequence diagram of record selection form operation

A UML sequence diagram outlining the interaction of programming objects during typical operations of a “record selection” form. This is perhaps the most complex programming in the application, with many interactions between custom VBA class modules. The sequence diagram can be helpful to a programmer seeking to understand the programming mechanisms at work.

6. Forms Navigation Model Diagram

This diagram provides a high-level flowchart representing the navigation paths from one form to another.

7. Source Code

VBA source code is provided in accessible form with the application.