

Modularization Plan

SelCtlDB V1 Development Project – Client Application

Larry Bednar

June 14, 2003

Purpose

This document describes code modularization used in this project.

[LB TO-DO

- Complete descriptions of selection fields to be provided on “record selection” forms – cross-check with client staff.
- Complete descriptions of list box data sources to be provided on “selected record display” forms. Determine whether alternate displays of columns are required in these list boxes (possible to provide alternatives via command buttons resetting the “row source” property).
- Complete descriptions of report types to be provided on “selected record display” forms. Need more detail on spatial layout, etc.
- Complete descriptions of local tables holding data for population of list boxes and combo boxes and complete the design of processes for downloading that data at time of client-side application start-up.
- Establish standard abbreviations for table names (to be used in table aliases, etc). Decide whether these will be placed in a table or globally accessible recordset, module-level user-defined type, custom object collection, etc. to be readily referenced during application operations, development work, etc. Some mechanism tightly linking table name with abbreviation may prevent errors resulting from the developer not using the same table name/alias in each “SelCtl...” object used in the form...(Required so that the constructed SelClause value for each “SelCtl...” references the correct column...). The use of a custom object collection probably is the most fool-proof and error-preventative approach, but the use of a user-defined type or multidimensional array would be faster in performance terms.
- Decide whether record entries should be treated as “transactions” not yet committed to the system until the user hits “save”. This is possible, but probably takes more effort than simply allowing Access to submit records for save at the usual default times.
- There is a need in many places for a modular tool to assist the user in selecting a single party from those tracked by the system. It seems this is a good opportunity for a separate global module using a class object with a number of standard controls defined as properties, etc. This would provide the user with the same selection system in all places they use it – should enhance usability.

- Most forms used by administrators (code edits, staff_mbr edits, etc.) are planned to be datasheet or list-style forms. A set of common features/buttons to be used for list-style forms should be developed and used to provide a richer, but still consistent and efficient UI for those editing functions. Datasheet is probably not really acceptable, even solely for administrators.
- Decide whether “code definition...” forms may all be implemented with a single form using a procedure that sets the data source appropriately.
- Decide whether “main record edit” and “related record edit” forms may all be implemented on a single, tabbed form.

]

Copyright/License

Copyright © 2004 Larry Francis Bednar

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Overview

The application will contain two major components: a “server application” where data resides on a permanent basis, and a “client application” which is installed on computers used by database workers to access and manipulate data held in the “server application”.

The server application will be constructed in MS-Access, MS SQL Server, or in MySQL.

The client application will be constructed using Microsoft Access 2000 or later using the ActiveX Data Objects (ADO) object model.

Naming Conventions

Naming conventions for data objects (tables, queries, views, columns, etc.) differ from those applied to MS-Access form controls, VisualBasic procedures, functions, classes, properties, etc.

Because of the amount of VisualBasic code to be generated, shorter names were found advantageous in code use. As a result, a somewhat different set of naming conventions was applied to objects that were to be referenced primarily in VisualBasic code. In general, VisualBasic names will use shorter word abbreviations, will delimit abbreviations/words by capitalizing the first letter (rather than using underscores delimiters), and will use the Reddick naming conventions as outlined in Appendix A of the Access 2000 Developer’s Handbook, Volume 1: Desktop Edition.

Form controls are treated primarily as VisualBasic code objects, with the same basic naming approach as used with VisualBasic variables, etc.

Module Descriptions

This section contains descriptions of code modules planned for use in the client application. Modules are described in terminology appropriate to MS-Access.

Forms

This section contains descriptions of all forms to be provided in the new system.

In general, all edit forms provided as part of the application will employ SQL statements assigned to the “data source” property. This approach will ensure that no queries, etc. are exposed to the user in a way that might allow alteration, etc. The user will therefore have the “query” area of the client application completely available for their own use in defining/storing queries.

All forms will reside in the “client” portion of the application, leaving choices of DBMS, etc. for the “data server” portion of the application the maximum possible flexibility.

Forms will derive all possible data used for populating combo boxes and list boxes from local tables populated by download from data server at start-up of client side application.

Form – Common “Record selection form” characteristics

Most supported tasks in the new system will use a “record selection form”. Many of the characteristics of these forms are common. Those characteristics held in common are described in this section.

In “data editing” tasks this tool will perform the following functions:

1. Allows the user to efficiently search for a record to be edited
2. Facilitates users’ systematic checking for pre-existing records prior to their entry of a possible duplicate record.
3. Reduces the amount of data passed between data server and client interface – facilitating high performance of the system by avoiding unnecessary data transfers across network.

In “reporting” tasks this tool will perform the following functions:

1. Allows the user to efficiently select the set of records to be represented in a report, and exported data file, etc.

Forms of this type will provide a variety of unbound controls (no bound controls will be used) facilitating the user’s specification of the most commonly used selection criteria. Provided controls will typically include:

1. “Start date” – used to specify a time period into which some stored date must fall
2. “End date” – used to specify a time period into which some stored date must fall
3. “Effective date” – used to specify a date on which a group membership must be in effect

4. “Ineffective date” – used to specify a date that must fall outside of any group membership period recorded in the system.
5. “text search pattern” – used to specify text patterns used to select records with matching values
6. “multi-select list” – used to allow the user to specify several categories, groups, etc. to which selected records must belong

There is a possibility that combo box and list box controls will be updated dynamically to provide value lists derived from the set of records identified by selection criteria already entered by the user. However, while this would be desirable from the user’s perspective, it also would require repeated queries of the data server with potentially large volumes of data passed to the form’s control. Each time the user updates a control, all displayed value lists, etc. must then be updated by query of the data server. No actual performance testing has yet been undertaken to verify performance degradation, but current plans are to specify static list box and combo box data sources to avoid this potential problem.

Present plans are that controls for these selection criteria will be declared to be members of a custom “selection control” class. This class will provide properties useful for the functions required by the selection form, including:

“ValueProvided” – Boolean “yes” if user has provided information using the control

“FilterClause” – An SQL clause that may be incorporated into an MS-Access filter clause to impose the condition indicated by the user.

[LB – The idea described in this paragraph is still “brainstorming” stage...] As the user updates each control, the form constructs a query to the database that returns only the count of records identified by the current selection criteria. This is done so that the user might be quickly informed if they have entered criteria that match no records currently in the system. Without this prompt response, the user may find it difficult to know which part of their specification must be altered to return *some* rows. The record count is selected because it is nearly the minimum amount of information required for this purpose. If a zero record count is returned: a message box is opened informing the user that the overall selection criteria they’ve entered match no records in the system, the last change made is reverted, and the prompt is returned to the last control updated by the user. If a non-zero count is returned, the count is displayed in a read-only control on the form for the user’s benefit. This information should be of assistance to a user who wants to know whether to add additional selection criteria to pare down their selection list further.

The form will also provide the following command buttons:

1. “Display matching records” – Typically, this moves the user to a related “selected records display form” where a concise display of information from records matching the user-specified selection criteria is displayed. The core query against the data server is not executed until this function is selected by the user. (Executes “OpenSelectedRecsForm” procedure).
2. “Exit” – Sends the user back to the main switchboard. (Executes “ExitRecSelectionForm” procedure.)
3. “New selection” – Deletes all selection criteria set by user and places them in a suitable position to start entering a new set of selection criteria.

The form will construct and SQL “filter” clause representing the user-specified selection criteria and will generally pass this clause to a “selected records display form” to enable that form to display identifying information for records matching the user-specific selection criteria.

Private module-level variables include:

mstrRecDisplayForm – references the associated “selected records display form”.

mstrBaseSelectStmt – provides the main body of the SQL SELECT statement used to derive row counts of selected records.

mstrSelFilter – contains text of Filter expression constructed to implement user’s selection control choices.

Standard private modules include:

ExitToMain – Exits form, returns user to main switchboard.

InitiateForm – Sets values of all properties for “SelCtl...” class instances appropriately, as well as their properties. Sets value of mstrSelFilter variable to blank. Define the base SQL “SELECT” statement to be used for record count queries to the server. Add “SelCtl...” class instances to custom collection used to easily construct mstrSelFilter value.

NewSelCriteria – Resets all selection criteria to blank. Resets value of selection filter to blank.

OpenSelectedRecsForm – Minimizes form, sends user to associated “Selected records display” form.

SetSelFilter – Called by “control update” events for all selection controls. Constructs filter expression used to implement user-specified choices. Each form may require a custom version of this module, or it may be possible to use collections of “SelCtl...” objects to create code that will work correctly on all forms...

SetSelRowCount – [LB – Still a “maybe”, pending a bit more investigation work on implementation mechanisms and performance effects...] Called by SetSelFilter procedure? Sends “row count” query to data server. If non-zero result, sets value of read-only control used to display this information for user. If zero result, displays window informing user, then returns user to last control updated and reverts the last change to that control.

Form – Common “Selected records display form” characteristics

Most supported tasks in the new system will employ a “selected records display” form. Many of the characteristics of these forms are common. Those characteristics held in common are described in this section.

This form provides the user a compact display of the records selected by their entries on the associated “record selection form”. All controls provided will be unbound. The display is provided in a list box control (**lstRecsDisplay**) using a data source property into which the filter clause constructed by the associated “record selection form” has been inserted.

In data editing tasks, this form provides the following functions:

1. Displays an abbreviated list of records matching characteristics entered by the user.

This facilitates the user's selection of a single record for editing. The amount of help provided depends on the skill of the user in specifying good selection criteria and the amount of information available for use in selection criteria. The user should not be presented a pull-down list of thousands of records to select the record to be edited. It is preferable that the user is required to examine a list of only a few records matching selection criteria they enter.

2. This display allows the user to quickly check whether a record representing the information of concern is pre-existing in the system. This should help reduce duplicate entries of information.
3. The **lstRecDisplay** control will be set up to allow a single record to be selected.

In reporting tasks, this form provides the following functions:

1. the user reviews the set of selected records to verify that they are the correct set of records to include in the report.
2. The user is provided the capacity to indicate the type of output to be generated, and the system produces the desired report.

The form will also provide a mechanism for the user to select a "report/output" method to be used in summarizing the selected records. The options provided will depend on the specific task being supported by the form. To preserve screen space and clarity, a combo box may be the best choice for displaying possible user selections. It may be desirable to alter the display of "selected records" according to the "report" option selected by the user.

The user will be allowed to select the following actions:

1. "Exit" – sends the user back to the main switch board
2. "Revise selection" – sends the user back to the associated "record selection form" to revise the user-specified selection criteria. (Executes "OpenRecSelectionForm" procedure.)
3. "Function" – option group or checkbox allowing the user to indicate whether "edit" or "report functions" are to be performed.
4. "Output format" – displays a list of the available output formats in a combo box, enabled only if the user has indicated "report" functions should be performed.
5. "Report" – performs the reporting action specified by the user (enabled only if the user has indicated "report" functions should be performed)
6. "Edit selected record" – Opens the record selected by the user for editing (enabled only when the user has indicated "edit" functions should be performed)
7. "Delete selected record" – Deletes the record selected in the lstRecDisplay list box. [LB – Is this safe? Or should the user be forced to view the full contents of the record using an Edit form before being allowed to delete?]

8. “New record” – Creates a new record and prompts the user for values (enabled only when the user has indicated “edit” functions should be performed)

Module level variables include:

mstrBaseSelectStmt – Provides main portion of SQL SELECT statement to be used as value of data source property for lstRecDisplay list box..

mstrSelectedRpt – Refers to report selected by user for use with displayed set of records

mstrRecSelForm – Refers to associated record selection form.

mstrSelFilter – Contains text of selection filter obtained from associated “record selection form”.

Standard private modules include:

CreateNewRec – Creates a new record and opens the associated “Edit” form.

DeleteRec – Deletes record selected in lstRecDisplay list box, refreshes lstRecDisplay.

EditRec – Opens the record selected by the user from the lstRecDisplay list box using the associated “Edit” form.

ExitToMain – Exits form, returns user to main switchboard.

InitiateForm –Sets value of row source property for lstRecDisplay list box.

NewRecSelection – Closes form, sends user to associated “selected records display” form with all criteria set to blank.

OpenReport – Opens the user-specified report using the current filter expression. Closes the minimized “record selection form” associated with this form.

ReviseRecSelection – Closes form, sends user to associated, minimized “Selected records display” form.

Form – Common “main record edit form” characteristics

Main data subjects in the system are accessed for editing using these forms. The “data edit form” by contrast allows the user to exit to a “selected record display” form. Several features are common to all forms supporting such functionality.

These forms generally display attributes of a single record in the subject area. Identifying information for the “parent” record is displayed in “read-only” controls.

Controls for all attributes allowing user-specification are provided.

In cases where editing of related data areas seems preferable to provision of subforms, etc the form will provide command buttons that allow the user to access related records associated with the displayed “main subject” data record. As an example, a “party data edit form” may provide commands buttons to open forms designed for editing of “locations”, “phones”, or “email” associated with the “party”.

The user will be allowed to select the following actions:

1. “Exit” – sends the user back to the main switch board
2. “Delete” deletes the selected record.
3. “Save” – Returns the user to the associated “selected records display” form, discarding changes or data entered by the user.
4. “Start anew” – discards changes already entered for the selected record, sets all values to their original values and allows the user to begin data editing again.

Module level variables include:

mstrBaseSelectStmt – Provides main portion of SQL SELECT statement to be used as value of data source property for lstSelectedRecsDisplay list box..

mstrRecDisplayForm – Refers to associated “selected records display form”.

mstrSelFilter – Contains text of selection filter obtained from associated “record selection form”.

Standard private modules include:

CreateNewRec – Creates a new record and opens the associated “Edit” form.

DeleteRec – Deletes the displayed record, returns user to associated, minimized “Selected records display” form.

InitiateForm –Sets value of form-level filter, data source, etc.

NewRecSelection – Closes form, sends user to associated “selected records display” form with all criteria set to blank.

OpenReport – Opens the user-specified report using the current filter expression. Closes the minimized “record selection form” associated with this form.

ReviseRecSelection – Closes form, sends user to associated, minimized “Selected records display” form.

Form – Common “Related data edit form” characteristics

Several data subjects in the system are accessed as “related data” from a main subject area. One important distinction between these forms and the “data edit form” are that the “related data edit form” allows the user to exit back to a “parent” record in the data area, and does not allow the user to exit directly to a “selected records display” form. Several features are common to all forms supporting such functionality.

These forms generally display a list of records in the related subject area that are related to a single record in the “parent” data area. Identifying information for the “parent” record is displayed in “read-only” controls.

A concise display of the records currently stored in the related data area is provided using a list box (lstSelectedRecsDisplay). The user may select any of the displayed records to edit values on that record, or may choose to create a new record. When the user selects a record, the values of that record are displayed on the screen for the user's modification.

The user will be allowed to select the following actions:

1. "Delete" deletes the selected record.
2. "Edit selected record" – Opens the record selected by the user for editing
3. "New record" – Creates a new record and prompts the user for values
4. "Return to parent form" – Returns the user to the "parent" form.
5. "Save" – saves the data changed by the user
6. "Start anew" – discards changes already entered for the selected record, sets all values to their original values and allows the user to begin data editing again.

Module level variables include:

BaseSelectStmt – Provides main portion of SQL SELECT statement to be used as value of data source property for lstSelectedRecsDisplay list box..

SelectedReport – Refers to report selected by user for use with displayed set of records

ParentForm – Refers to associated parent form (the form for the main data subject from which this form was reached).

SelFilter – Contains text of selection filter obtained from associated "record selection form".

Standard private modules include:

CreateNewRec – Creates a new record and opens the associated "Edit" form.

EditSelectedRec – Opens the record selected by the user using the list box using the associated "Edit" form.

ExitToParent – Exits form, returns user to parent form.

InitiateForm –Sets value of SelFilter, etc.

OpenReport – Opens the user-specified report using the current filter expression. Closes the minimized "record selection form" associated with this form.

Form – Admin Setup Switchboard

Provides the user the option of switching to any of the "code definition edit" forms. Intended for use by administrators only.

| Control Name | Type | Remarks |
|--------------------------|----------------|--|
| cmdEditContactMethod | Command button | Displays the related “code definition edit form” |
| cmdEditContactSubj | Command button | Displays the related “code definition edit form” |
| cmdEditCounty | Command button | Displays the related “code definition edit form” |
| cmdEditDonType | Command button | Displays the related “code definition edit form” |
| cmdEditFund | Command button | Displays the related “code definition edit form” |
| cmdEditGrp | Command button | Displays the “group edit” form |
| cmdEditGrpType | Command button | Displays the related “code definition edit form” |
| cmdEditLoctnType | Command button | Displays the related “code definition edit form” |
| cmdEditPartyCat | Command button | Displays the related “code definition edit form” |
| cmdEditPhone | Command button | Displays the related “code definition edit form” |
| cmdEditRelatedPersonRole | Command button | Displays the related “code definition edit form” |
| cmdEditRgn | Command button | Displays the related “code definition edit form” |
| cmdEditStaffMbr | Command button | Displays the “staff mbr edit” form |

Form – Code Definition Edit Contact Method

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Contact Subject

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit County

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Donation Type

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Fund

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Group Type

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Location Type

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Party Category

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Phone Type

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Region

Datasheet style form for editing records, for use only by administrators.

Form – Code Definition Edit Related Person Role

Datasheet style form for editing records, for use only by administrators.

Form – Contact Edit

Provides the user a display to assist in entry of information characterizing a contact.

Displayed controls include

| Control Name | Type | Remarks |
|-------------------|--------------------------------------|---|
| cboParty | Combo box | Bound to party_id column. 1 st column is party_id, 2d column is party.name. May require an additional control to restrict the set of values displayed in the combo box to a manageable number. |
| cboStaffMbr | Combo box | Bound to staff_mbr_id column. Row source local_staff_mbr table. 1 st column is staff_mbr_id, 2d column is staff_mbr.name |
| lstContactSubject | Multi-select list box?? Subform?? | Bound to contact_subject_defn_id column. Underlying procedure used to add records to related table when user selects to save entries. |
| txtContactDate | Text box, date format | Bound to contact_date column. |
| txtContactID | Text box, counting number format | Bound to contact_id column. |
| txtContactMethod | List box, single select | Bound to contact_method_id column. |
| txtRemk | Text box | Bound to remk column. |

[LB – It might be desirable to simply provide these controls on the same form as the “selected record display form”. This would eliminate the need for the user to navigate to a different form. The user’s selection of a record from the list box displaying selected records could result in display of the information items on the bottom portion of the form. The issue of whether this is a desirable approach depends partly on how many records the user will typically display in that list box. If a large number of records is typical, the list box will require so much space on the “selected record display form” that inadequate space will be available for display of values from an individual record. Also, this approach will definitely *not* be useable in some other subject areas where the number of characteristics will simply be too large – it might be preferable to keep the user within exactly the same pattern of form movement in *all* data areas...]

The form will provide the same commands outlined above in the section ‘Form – Common “main record edit form” characteristics’.

Form – Contact Record Selection

Provides the user a number of unbound controls to be used in selecting a subset of contact records stored in the system. This is intended to facilitate the user’s selection of a record to be edited, or to verify that a new record to be entered does not duplicate a pre-existing record in the system.

The basic characteristics are as described above (‘Form – Common “record selection form” characteristics’).

Unbound controls to be provided for selection purposes include:

| Control Name | Type | Custom Class | Remarks |
|---|--------------------------------------|---|---|
| chkContactMethodInteract | Check box? Option group? | | Sets lstContactMethod InteractType property of SelCtlMultLst |
| chkContactSubjectInteract | Option group? | | Sets lstContactSubject InteractType property of SelCtlMultList object |
| lstContactMethod | Multi-select list box | SelCtlMultLst | Specifies search conditions targeted to Contact.contact_method_id column |
| lstContactSubject | Multi-select list box | SelCtlMultLst | Specifies search conditions for Contact.subject.contact_subject_id column |
| lstStaffMbr | Multi-select list box | SelCtlMultLst, InteractType property set to “any selection” | Specifies search conditions for Contact.staff_mbr_id, column |
| txtDtMax | Text box, date format | SelCtlIntervalEnd | Specifies search conditions for Contact.contact_date column |
| txtDtMin | Text box, date format | SelCtlIntervalEnd | Searches Contact.contact_date column |
| txtPartyNamePattern | Text box | SelCtlTxtPattern | Specifies search conditions for Party.Party.name column |
| txtSelRecordCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Set by “control update” event procedure to display count of records selected by user’s choices. |
| | | | |
| | | | |

The SetSelFilter procedure of the form will be customized to correctly combine values obtained from the selection controls.

Form – Contact Selected Records Display

Provides the user a compact display of information for records matching the criteria entered in the associated “record selection” form.

Controls to be provided include:

| Control Name | Type | Remarks |
|-------------------|---------------------------------|---|
| lstRecDisplay | List Box, single-selection only | Displays the following columns from selected rows of Contacts table or closely related tables: contact_id, staff_mbr.name, contact_date, contact_method.abbr, contact_subject.abbr, remks (approx. 1 st 20 characters) |
| txtSelDescDisplay | Text box | Unbound, Displays “human-language” version of filter used to select records displayed in lstRecDisplay list box |

The user will be provided the standard function/command choices outlined under ‘Form – Common “selected record display form” characteristics’ above.

Form – Donation Edit

Provides the user a display to assist in entry of information characterizing a donation record..

Displayed controls include

| Control Name | Type | Remarks |
|------------------|--------------------------------------|---|
| chkProxyDonYN | Check box | Bound to proxy_donation_yn column. |
| lstDonType | List box, single selection only | Bound to donation_type_defn_id column. Row source local_donation_type table, ordered by abbr, 1 st column hidden and containing donation_type_id values, 2d column displayed and containing abbr values. |
| lstFinalDonor | List box, read only | Bound to final_donor_party_id column. May require tools to assist user in searching for correct party. Displays enough columns to clearly identify the selected party. Set to match selection in sfrFinalDonorSel control. |
| lstFund | List box, single selection only | Bound to fund_defn_id column. Row source local_fund_defn table, ordered by abbr, 1 st column hidden and containing fund_id values, 2d column displayed and containing abbr values |
| lstOrigDonor | List box, read only | Bound to orig_donor_party_id column. Enabled only if “chkProxyDonYN” is set to “yes”. Displays enough columns to clearly identify the selected party. Set to match selection in sfrOrigDonorSel control. |
| lstPledge?? | List box??? | Bound to pledge_yn column. [LB – This control requires some more thought – may need to provide a mechanism to assist the user in selecting from numerous pledge records.] |
| sfrFinalDonorSel | Subform | Provides search tools to assist user in selecting a single party to assign as “final donor” |
| sfrOrigDonorSel | Subform | Provides search tools to assist user in selecting a single party to assign as “orig donor” Enabled only if “chkProxyDonYN” is set to “yes”. |
| txtAmt | Text box, currency format | Bound to amt_provided column. Non-negative values only, form-based validation, enabled property set to “No” if donation type is in (“service”, “in-kind”,...) |
| txtDonDt | Text box, date format | Bound to donation_date column. [LB – Should this be form-validated? No greater than today? No less than some much earlier date?] |
| txtDonID | Text box, counting number, read-only | Bound to donation_id column. |

| Control Name | Type | Remarks |
|--------------|---------------------------|---|
| txtRemk | Text box | Bound to remk column. |
| txtValue | Text box, currency format | Bound to value column. Non-negative values only, form-based validation, enabled property set to “No” if donation type is not in (“service”, “in-kind”,...) |

[LB – Is there a need to display information about a related pledge record? Pledge date, etc? Note that the pledge is associated with a “party” as is the “donation” itself. There may be an opportunity to streamline data entry by not requiring the user to enter a link to a “party” in both donation and pledge...]

The form will provide the same commands outlined above in the section ‘Form – Common “main record edit form” characteristics’.

Form – Donation Record Selection

Provides the user a number of unbound controls to be used in selecting a subset of donation records stored in the system, to facilitate the user’s selection of a record to be edited, or to verify that a new record to be entered does not duplicate a pre-existing record in the system.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|--------------|---------------------------------|-------------------|--|
| lstDonType | Multi-select list box | SelCtlMultLst | Used to search values of donation_type_id column. Row source from local_donation_type table, 1 st column displays donation_type_id, 2d column displays abbr |
| lstFund | Multi-select list box | SelCtlMultLst | Used to search values of fund_id column. Row source from local_fund_defn table, 1 st column displays fund_id, 2d column displays abbr |
| lstProxyDon | List box, single selection only | SelCtlMultLst??? | Used to search values of proxy_donation_yn column. Possible values “include”, “exclude”, “only”. [LB – Will this custom class object work for this user input?] |
| txtDonAmtMax | Text box, currency format | SelCtlIntervalEnd | Used to search values of amt_provided column. Form-based validation to ensure that the entered value is a valid date >= txtDonAmtMin. |
| txtDonAmtMin | Text box, currency format | SelCtlIntervalEnd | Used to search values of amt_provided column. Form-based validation to ensure that the entered value is a valid non-negative value <= txtDonAmtMax. |
| txtDonDtMax | Text box, short date format | SelCtlIntervalEnd | Used to search values of donation_date column. Form-based validation to ensure that the entered value is a valid date >= txtDonDtSt. |
| txtDonDtMin | Text box, short date format | SelCtlIntervalEnd | Used to search values of donation_date column. Form-based validation to ensure that the entered value is a valid date <= txtDonDtEnd. |
| txtDonValMax | Text box, | SelCtlIntervalEnd | Used to search values of value column. Form-based |

| Control Name | Type | Custom Class | Remarks |
|---|--------------------------------------|-------------------|---|
| | currency format | | validation to ensure that the entered value is a valid date >= txtDonValMin. [LB – “OR” combination with donation.amt is implied] |
| txtDonValMin | Text box, currency format | SelCtlIntervalEnd | Used to search values of value column. Form-based validation to ensure that the entered value is a valid non-negative value <= txtDonValMax. [LB – “OR” combination with donation.amt is implied] |
| txtPartyNameSearch | Text box | SelCtlTxtSearch | Used to search values of party.name column. [LB – Need some decisions about how to deal with single/double quotes. Current decision is to convert to single-character wild cards in search pattern specification for SQL “LIKE” operator.] |
| txtPersonNameSearch | Text box | SelCtlTxtSearch | Used to search values of person.name column. [LB – Need some decisions about how to deal with single/double quotes. Current decision is to convert to single-character wild cards in search pattern specification for SQL “LIKE” operator.] |
| txtSelRecordCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Set by “control update” event procedure to display count of records selected by user’s choices. |

The user is provided the command options outlined in the section ‘Form – Common “Record Selection Form” Characteristics’ above.

Form – Donation Selected Record Display

Provides the user a compact display of information for records matching the criteria entered in the associated “record selection” form.

Controls to be provided include:

| Control Name | Type | Remarks |
|-------------------|---------------------------------|--|
| lstRecDisplay | List box, single selection only | Displays values stored in columns of the donation table and closely related tables. Columns displayed include: donation_id, party.name, donation_type.abbr, fund.abbr, donation_date, amt, value, proxy_donation_yn, party.name (through “original donor” relationship). |
| txtSelDescDisplay | Text box, read-only | Displays the “human-language” version of the filter expression used to select the records displayed in the lstRecDisplay list box. |

The user will be provided the standard function/command choices outlined under ‘Form – Common “selected record display form” characteristics’ above.

Form – Easement Project Edit

Provides the user a display to assist in entry of information characterizing a ??.

Displayed controls include

| Control Name | Type | Remarks |
|---------------|-------------------------------------|--|
| lstRgn | List box | Bound to region_defn_id column. Row source local_region_defn table. 1 st column hidden, displays region_id. 2d column displays region.abbr. |
| lstParties | List box, read-only | Displays related “easement project party” records. |
| sfrStaff | Subform, datasheet type?? | Number of staff members is relatively small – a straight forward subform approach displaying related records in easement_proj_staff table should be adequate. [LB – subform will need to be defined if final decision is to use this approach.] |
| txtAbbr | Text box | Bound to abbr column. |
| txtEmntProjID | Text box, counting number, read-onl | Bound to easement_proj_id column |
| txtPropName | Text box | Bound to property_name column |
| txtRemk | Text box | Bound to remk column. |
| | | |

The form will provide the same commands outlined above in the section ‘Form – Common “main record edit form” characteristics’.

[LB – If a desire for a more refined ability to cancel a complete entry until the last minute is desired, a multi-select list box could be programmed for use in place of subStaff. However, similar replacement of subParties may be more difficult.]

Form – Easement Project Record Selection

Provides the user a number of unbound controls to be used in selecting a subset of easement project records stored in the system, to facilitate the user’s selection of a record to be edited, or to verify that a new record to be entered does not duplicate a pre-existing record in the system.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|----------------|-----------------------|---------------|--|
| lstLandSteward | Multi-select list box | SelCtlMultLst | Used to search for related proj_staff records with role value of “land steward” and staff_mbr_id as indicated by the user selection. Row source local_staff_mbr table. 1 st column hidden, holds staff_mbr_id. 2d column displays name. Rows ordered by name. |
| lstMstoneType | Multi-select list | SelCtlMultLst | Used to search values of easement_proj_milestone.type |

| Control Name | Type | Custom Class | Remarks |
|---|--------------------------------------|-------------------|--|
| | box | | column |
| lstProjMgr | Multi-select list box | SelCtlMultLst | Used to search for related proj_staff records with role value of “proj mgr” and staff_mbr_id as indicated by the user selection. Row source local_staff_mbr table. 1 st column hidden, holds staff_mbr_id. 2d column displays name. Rows ordered by name. |
| lstRgn | Multi-select list box | SelCtlMultLst | Used to search values of region_id column. Row source local_region_defn table. 1 st column hidden, holds region_id. 2d column displays name. Rows ordered by name. |
| txtMstoneDtMax | Text box, short date format | SelCtlIntervalEnd | Used to search for range of values in milestone_date column |
| txtMstoneDtMin | Text box, short date format | SelCtlIntervalEnd | Used to search for range of values in milestone_date column |
| txtPartyName | Text box | SelCtlTxtSearch | Used to search for related proj_party records linked to party records with name column matching the user-specified search pattern. |
| txtPropName | Text box | SelCtlTxtSearch | Used to search values of property_name column. |
| txtSelRecordCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Set by “control update” event procedure to display count of records selected by user’s choices. |
| | | | |

The user is provided the command options outlined in the section ‘Form – Common “Record Selection Form” Characteristics’ above.

Form – Easement Project Selected Record Display

Provides the user a compact display of information for records matching the criteria entered in the associated “record selection” form.

Controls to be provided include:

| Control Name | Type | Remarks |
|-------------------|---------------------------------|--|
| lstRecDisplay | List box, single selection only | Displays values stored in columns of the easement_project table and closely related tables. Columns displayed include: abbr, prop_name (1 st 20 characters), land steward (staff_mbr.name through easement_proj_staff association with role of “land steward”), project manager (staff_mbr.name through easement_proj_staff association with role of “proj mgr”), last milestone (easement_proj_milestone.type value from associated record with greatest milestone_date value) |
| txtSelDescDisplay | Text box, read-only | Displays the “human-language” version of the filter expression used to select the records displayed in the lstRecDisplay list box. |

The user will be provided the standard function/command choices outlined under ‘Form – Common “selected record display form” characteristics’ above.

Form – Group Edit

Provides the user a display to assist in entry of information characterizing a group. [LB – Could this simply be a datasheet display or similar? The number of columns and groups is likely to be small enough for this to be pretty effective for the user. In addition, it seems like it would be best if update of this data is restricted to the administrator group. So a finely honed UI may be unnecessary.]

Displayed controls include

| Control Name | Type | Remarks |
|--------------|---|--|
| lstGrpType | List box | Bound to group_type_id column. Row source local_group_type_defn table. 1 st column hidden, holds group_type id. 2d column displays abbr. Rows ordered by abbr. |
| txtAbbr | Text box | Bound to abbr column. Form-level validation to ensure each record has a unique abbreviation? |
| txtDefn | Text box | Bound to defn column. |
| txtGroupId | Text box, counting number format, read-only | Bound to group_id column. |
| txtRemk | Text box | Bound to remk column. |
| | | |

The user is provided the command options described in the section ‘Form – Common “main record edit form” Characteristics’ above. [LB – This won’t be quite right, as there will be no “record selection form” preceding this form in process steps.]

Form – Main Switchboard

Provides user the option of switching to any of the “record selection” forms in the system, or to the “code definition switchboard” form.

| Control Name | Type | Remarks |
|----------------------|----------------|---|
| cmdAdminSetup | Command button | Displays the “admin setup switchboard” form |
| cmdContactSelection | Command button | Displays the “contact record selection” form |
| cmdDonationSelection | Command button | Displays the “donation record selection” form |
| cmdEasementProj | Command button | Displays the “easement project record selection” form |
| cmdPartySelection | Command button | Displays the “party record selection” form |

| Control Name | Type | Remarks |
|--------------------|----------------|---|
| cmdPersonSelection | Command button | Displays the “person record selection” form |
| cmdPledge | Command button | Displays the “pledge record selection” form |
| | | |
| | | |

Form – Party Edit

Provides the user a display to assist in entry of information characterizing a ??.

Displayed controls include

| Control Name | Type | Remarks |
|---------------------|--|---|
| cmdEditEmail | Command button | Opens associated “related record edit form”. Passes filter to limit display to records related to this “party” record. |
| cmdEditGrpMbrPeriod | Command button | Opens associated “related record edit form”. Passes filter to limit display to records related to this “party” record. |
| cmdEditLoctn | Command button | Opens associated “related record edit form”. Passes filter to limit display to records related to this “party” record. |
| cmdEditPartyMbr | Command button | Opens associated “related record edit form”. Passes filter to limit display to records related to this “party” record. |
| cmdEditPhone | Command button | Opens associated “related record edit form”. Passes filter to limit display to records related to this “party” record. |
| lstPartyType | List box | Bound to type column. Row source?? |
| lstStaffMbr | List box | Bound to staff_mbr_id column. Row source local_staff_mbr. 1 st column hidden, holds staff_mbr_id. 2d column displays abbr. Rows ordered by abbr. |
| sfrPartyCat | Multi-select list box?? Datasheet subform?? | Displays related records in party_category table. Some need for validation to prevent repeated entry of the same category code. [LB – multi-select list box probably provides greater power for validation of entries, etc. but also will require more programming/expense] |
| txtPartyId | Text box, counting number, read-only | Bound to party_id column |
| txtPartyName | Text box | Bound to name column. |

The form will provide the same commands outlined above in the section ‘Form – Common “main record edit form” characteristics’.

Form – Party Edit Related Email

Provides the user a display to assist in entry of information characterizing a email record associated with a “party” record that was being viewed using the Party Edit form when the user hit the command button to move to this form.

Displayed controls include

| Control Name | Type | Remarks |
|---------------|---|--|
| chkPrimaryYN | Check box?? Radio buttons?? | Bound to primary_yn column |
| lstRecDisplay | List Box | Displays concise description of each email record associated with parent “party” record. Row source is an equi-join of email with email_type_defn and party, using a filter passed from the parent form to ensure that only phone records associated with the parent “party” record are displayed. |
| txtAddress | Text box | Bound to address column. [LB – Is there a need for validation of any sort?] |
| txtEmailId | Text box, counting number, read-only | Bound to email_id column |
| txtPartyID | Text box, counting number format, read-only | Displays info identifying “parent” party for user benefit. |
| txtPartyName | Text box, read-only | Displays info identifying “parent” party for user benefit. |
| txtRemk | Text box | Bound to remk column. |

The user is provided the command options described in the section ‘Form – Common “Related record edit form” Characteristics’ above.

Form – Party Edit Related Group Membership Period

Provides the user a display to assist in entry of information characterizing a group membership period.

Displayed controls include

| Control Name | Type | Remarks |
|-------------------|---|--|
| lstRecDisplay | List Box | Displays concise description of each groupr_mbr_period record associated with parent “party” record. Row source is an equi-join of group_mbr_period with group, group_type, and party, using a filter passed from the parent form to ensure that only phone records associated with the parent “party” record are displayed. |
| txtDtEnd | Text box, short date format | Bound to date_end column. Form-level validation to ensure >= txtDtStart |
| txtDtStart | Text box, short date format | Bound to date_start column. Form-level validation to ensure <= txtDtEnd. |
| txtGrp | Combo box | Bound to group_id column. Row source local_group_defn table. 1 st column hidden holds group_id. 2 ^d column displays abbr. Rows ordered by abbr. |
| txtGrpMbrPeriodId | Text box, counting number format, read-only | Bound to group_mbr_period_id column. |

| Control Name | Type | Remarks |
|--------------|---|--|
| txtPartyID | Text box, counting number format, read-only | Displays info identifying “parent” party for user benefit. |
| txtPartyName | Text box, read-only | Displays info identifying “parent” party for user benefit. |
| txtRemk | Text box | Bound to remk column. |

The user is provided the command options described in the section ‘Form – Common “Related record edit form” Characteristics’ above.

Form – Party Edit Related Loctn

Provides the user a display to assist in entry of information characterizing a location.

Displayed controls include

| Control Name | Type | Remarks |
|---------------|---|--|
| cboCounty | Combo box | Bound to county_defn_id column. Row source local_cnty_defn table. 1 st col hidden, holds cnty_id. 2d column displays name or abbr. Row sorted by name or abbr (whichever is displayed in 2d column). |
| cboRgn | Combo box | Bound to region_defn_id column. Row source local_region_defn table. 1 st col hidden, holds region_id. 2d column displays abbr. Rows sorted by abbr. |
| cboState | Combo box | Bound to state column. Row source ????. One column displaying state abbreviations. Rows ordered by abbrev. [LB – Should the full names be displayed as well as the abbreviations?] |
| chkPrimaryYN | Check box?? Option group?? | Bound to primary_yn column. A value should always be entered, either “yes” or “no”. Form-level validation to ensure that only one location for each party is designated with a “yes”. Default “yes” will save keypunching effort, since most parties will have only one address, but will cause frequent exceptions when user enters multiple records and forgets to set additional records to “no”. It may be possible to set the default based on whether any other record has already been set to “yes”. |
| lstRecDisplay | List Box | Displays concise description of each location record associated with parent “party” record. Row source is an equi-join of location with region, county and party, using a filter passed from the parent form to ensure that only phone records associated with the parent “party” record are displayed. |
| txtAddress1 | Text box | Bound to address 1 column. |
| txtAddress2 | Text box | Bound to address 2 column. |
| txtCity | Text box | Bound to city column. |
| txtCountry | Text box | Bound to country column. [LB – Should this be a combo or list box?] |
| txtLoctnId | Text box, counting number format, read- | Bound to loctn_id column. |

| Control Name | Type | Remarks |
|--------------|---|--|
| | only | |
| txtPartyID | Text box, counting number format, read-only | Displays info identifying “parent” party for user benefit. |
| txtPartyName | Text box, read-only | Displays info identifying “parent” party for user benefit. |

The user is provided the command options described in the section ‘Form – Common “Related record edit form” Characteristics’ above.

Form – Party Edit Related Member

Provides the user a display to assist in entry of information characterizing a ??.

Displayed controls include

| Control Name | Type | Remarks |
|---------------|---|--|
| cboPerson | Combo box | Bound to related_person_person_id column. Row source ???. |
| chkPrimaryYN | Check box?? Option group?? | Bound to primary_yn column. A value should always be entered, either “yes” or “no”. Form-level validation to ensure that only one person for each party is designated with a “yes”. Default “yes” will save keypunching effort, since most parties will have only one person, but will cause frequent exceptions when user enters multiple persons as party members and forgets to set additional records to “no”. It may be possible to set the default based on whether any other record has already been set to “yes”. |
| lstRecDisplay | List box | Displays concise description of each party_mbr record associated with the parent “party” record. Row source is an equi-join of party_mbr, party and person, using a filter passed from the parent form to ensure that only records associated with the parent “party” record are displayed. |
| lstRole | List box | Bound to related_person_role_defn_id column. Row source ???. |
| txtPartyID | Text box, counting number format, read-only | Displays info identifying “parent” party for user benefit. |
| txtPartyName | Text box, read-only | Displays info identifying “parent” party for user benefit. |
| txtRemk | Text box | Bound to remk column. |

The user is provided the command options described in the section ‘Form – Common “Related record edit form” Characteristics’ above.

Form – Party Edit Related Phone

Provides the user a display to assist in entry of information characterizing a ??.

Displayed controls include

| Control Name | Type | Remarks |
|----------------|---|--|
| chkPrimaryYN | Check box?? Option box?? | Bound to primary_yn column. A value should always be entered, either “yes” or “no”. Form-level validation to ensure that only one phone for each party is designated with a “yes”. Default “yes” will save keypunching effort, since most parties will have only one phone, but will cause frequent exceptions when user enters multiple phones and forgets to set additional records to “no”. It may be possible to set the default based on whether any other record has already been set to “yes”. |
| lstRecDisplay | List box | Row source is an equi-join of phone with party and location, using a filter passed from the parent form to ensure that only phone records associated with the parent “party” record are displayed. Displayed columns include complete phone number, phone type, location type for phone records linked to location records. |
| lstType | List box, single selection only | Bound to phone_type_defn_id column. Row source local_phone_type table. 1 st row hidden, holds phone_type_id. 2d row displays abbr. Rows sorted by abbr. |
| txtAreaCode | Text box, counting number format | Bound to area_code column. [LB – Should this be a combo box or list box instead?] |
| txtCountryCode | Text box, counting number format | Bound to country_code column. |
| txtExt | Text box, counting number format | Bound to ext column. |
| txtNbr | Text box, phone number mask | Bound to nbr column. |
| txtPartyID | Text box, counting number format, read-only | Displays info identifying “parent” party for user benefit. |
| txtPartyName | Text box, read-only | Displays info identifying “parent” party for user benefit. |
| txtPhoneID | Text box, counting number format, read-only | Bound to phone_id column. |

The user is provided the command options described in the section ‘Form – Common “Edit form” Characteristics’ above.

Form – Party Record Selection

Provides the user a number of unbound controls to be used in selecting a subset of party records stored in the system, to facilitate the user's selection of a record to be edited, or to verify that a new record to be entered does not duplicate an pre-existing record in the system.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|--|--------------------------------------|-----------------|---|
| lstCnty | Multi-select list box | SelCtlMultLst | Used to search values of column locn.cnty_id. Row source local_cnty_defn table. 1 st column hidden, holds cnty_id values, 2d column displays cnty.abbr. Rows ordered by cnty.abbr. Default “OR” use of multiple selections. |
| lstLocnType | Multi-select list box | SelCtlMultLst | Default “OR” use of multiple selections. [LB – Is there a need for an ability to choose an “AND” use of multiple selection?] |
| lstPartyCat | Multi-select list box | SelCtlMultLst | Used to search for associated records of party_category table. Default “OR” use of multiple selections. Row source local_party_category_defn table. 1 st column hidden, holds party_category.party_category_id values. 2d column displays party_category_defn.abbr values. Rows ordered by party_category_defn.abbr. [LB – Is there a need for an ability to choose an “AND” use of multiple selection?] |
| lstRgn | Multi-select list box | SelCtlMultLst | Used to search values of column locn.region_id. Row source local_region_defn table. 1 st column hidden, holds region_id. 2d column displays abbr. Rows ordered by abbr. Default “OR” use of multiple selections. |
| lstState | Multi-select list box?? | SelCtlMultLst | Used to search values of column locn.state. Row source local_st_defn table. 1 st column hidden, displays “standard” state abbreviations. Rows ordered by abbr. Default “OR” use of multiple selections. |
| txtCity | Text box | SelCtlTxtSearch | Used to search values of column locn.city. |
| txtEmail | Text box | SelCtlTxtSearch | Used to search values of column email.address. |
| txtPartyName | Text box | SelCtlTxtSearch | Used to search name column |
| txtSelRowCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Set by “control update” event procedure to display count of records selected by user's choices. |
| | | | |
| | | | |

The user is provided the command options outlined in the section ‘Form – Common “Record Selection Form” Characteristics’ above.

Form – Party Record Selection Subform

Provides a tool to assist in identification of a single party during editing of a “contact” record, a “donation”, an “easement project party”, etc. While similar to the “party record selection” form above, this form is designed for use as a control on a “parent” form and does not include command buttons linking to “edit” forms, etc. – the purpose is solely to assist the user to select a single “person record” and make a representation of that selection available to the “parent” form. Because of this, a number of properties and methods are defined that are similar to those defined for “SelCtl...” objects.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|--|--------------------------------------|-----------------|---|
| cmdUpdateDisplay | Command button | | Unbound control. Updates display in lstRecDisplay control to reflect user-entered selection criteria. |
| lstRecDisplay | List box, single selection only | | Unbound control. Row source is a subset of party table rows. Provides a compact display of party records and related information matching the search criteria entered in other controls on this subform. No default row source – row source is only set in response to user entry of selection criteria – this feature is designed to minimize data transfer from server to client. User selection fills SelClause property of subform. |
| txtNameSearch | Text box | SelCtlSearchTxt | Unbound control. Used by the user to enter text search pattern used to search party.name column values to select records for display in associated list box. Wildcards required. Apostrophes removed from search criteria to prevent run-time errors in execution of SQL LIKE operators. |
| lstType | list box, single selection only | SelCtlSearchTxt | Unbound control. Value selected by the user is used to search party.type column values. Wildcards required. Apostrophes removed from search criteria to prevent run-time errors in execution of SQL LIKE operators. |
| txtSelRowCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Unbound control. Set by “control update” event procedure to display count of records selected by user’s choices. |
| | | | |
| | | | |

[LB – Additional identification criteria may be required...]

Module-level variables

mstrSQLBase – portion of SELECT phrase for lstRecDisplay control row source specifying columns to be displayed.

mstrSQLOrder – ORDER clause for lstRecDisplay control row source .

mstrSQLWhere – WHERE clause for lstRecDisplay control row source .

Custom Properties

SelClause – Contains SQL WHERE clause phrase reflecting the user’s selection from the lstRecDisplay control.

Custom Methods

SetRecDisplayRowSource – Constructs SQL statement corresponding to user-entered selection criteria and sets the RowSource property of lstRecDisplay to that value.

SetSelClause – Set value of SelClause property to hold appropriate SQL WHERE clause phrase reflecting user selection from lstRecDisplay control. Read-only.

SetSelRowCount – [LB – Still a “maybe”, pending a bit more investigation work on implementation mechanisms and performance effects...] Called by SetSelFilter procedure? Sends “row count” query to data server. If non-zero result, sets value of read-only control used to display this information for user. If zero result, displays window informing user, then returns user to last control updated and reverts the last change to that control.

Form – Party Selected Record Display

Provides the user a compact display of information for records matching the criteria entered in the associated “record selection” form.

Controls to be provided include:

| Control Name | Type | Remarks |
|-------------------|---------------------------------|--|
| lstRecDisplay | List box, single selection only | Displays values stored in columns of the ?? table and closely related tables. Columns displayed include: add_date, name, type, staff_mbr.name, (other?? Primary location city/state, etc?) |
| txtSelDescDisplay | Text box, read-only | Displays the “human-language” version of the filter expression used to select the records displayed in the lstSelectedRecDisplay list box. |

The user will be provided the standard function/command choices outlined under ‘Form – Common “selected record display form” characteristics’ above.

Form – Person Edit

Provides the user a display to assist in entry of information characterizing a person record.

Displayed controls include

| Control Name | Type | Remarks |
|----------------------|---|--|
| cmdEditPersonEvent | Command button | Opens associated “related record edit form”, passes filter to limit displayed records in that form to those associated with the currently displayed record in this form. |
| cmdEditRelatedPerson | Command button | Opens associated “related record edit form”, passes filter to limit displayed records in that form to those associated with the currently displayed record in this form. |
| txtNameFirst | Text box | Bound to name first |
| txtNameLast | Text box | Bound to name last |
| txtNameMiddle | Text box | Bound to name middle |
| txtNamePrefix | Text box | Bound to name prefix |
| txtNameSuffix | Text box | Bound to name suffix |
| txtPersonId | Text box, counting number format, read-only | Bound to person_id |
| txtRemk | Text box | Bound to remk column |

The form will provide the same commands outlined above in the section ‘Form – Common “main record edit form” characteristics’.

[LB – Is there a need for additional identifying information to distinguish persons? Birthdate? State of birth? Other?]

Form – Person Edit Related Event

Provides the user a display to assist in entry of information characterizing a person_event record. Associated with the “parent” person record.

Displayed controls include

| Control Name | Type | Remarks |
|---------------|---|--|
| lstRecDisplay | List box, single select | Row source an equi-join of person, person_event, event_defn, filtered to show only those records related to the “parent” person record. |
| lstType | List box, sinble-select | Bound to person_event_type_defn_id column. Row source local_event_defn. 1 st column hidden, holds event_id, 2d column displays abbr. Rows ordered by abbr. |
| txtEventDt | Text box, short date format | Bound to event_date column. Default “today” value? |
| txtPersonID | Text box, counting number format, read-only | Displays person_id value of “parent person record. |
| txtPersonName | Text box, read-only | Ideally, should display full name of “parent” person record. |
| txtRemk | Text box | Bound to remk column |

The user is provided the command options described in the section ‘Form – Common “Related record edit form” Characteristics’ above.

[LB – It might be possible to simply combine this form with the “person edit” form by use of a subform on that display.]

Form – Person Edit Related Person

Provides the user a display to assist in entry of information characterizing a related_person record associated with the “parent” person record. .

Displayed controls include

| Control Name | Type | Remarks |
|--------------------|---|---|
| cboRelatedPerson | Combo box??? | Bound to related_person_person_id column. [LB – The user will probably need some type of assisting mechanism to easily choose a single person from among all those tracked by the system.] |
| lstRecDisplay | List box, single select | Row source an equi-join of person, person_event, event_defn, filtered to show only those records related to the “parent” person record. |
| txtPersonID | Text box, counting number format, read-only | Displays person_id value of “parent” person record. |
| txtPersonName | Text box, read-only | Ideally, should display full name of “parent” person record. |
| txtRelatedPersonId | Text box, | Bound to related_person_id column |
| txtRemk | Text box | Bound to remk column. |

The user is provided the command options described in the section ‘Form – Common “Related record edit form” Characteristics’ above.

[LB – It might be possible to simply combine this form with the “person edit” form by use of a subform on that display.]

Form – Person Record Selection

Provides the user a number of unbound controls to be used in selecting a subset of person records stored in the system, to facilitate the user’s selection of a record to be edited, or to verify that a new record to be entered does not duplicate an pre-existing record in the system.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|--|--------------------------------------|-------------------|---|
| txtNameFirst | Text box | SelCtlSearchTxt | Used to search values of name_first column |
| txtNameLast | Text box | SelCtlSearchTxt | Used to search values of name_last column |
| txtNameMiddle | Text box | SelCtlSearchTxt | Used to search values of name_middle column |
| txtNamePrefix | Text box | SelCtlSearchTxt | Used to search values of name_prefix column |
| txtNameSuffix | Text box | SelCtlSearchTxt | Used to search values of name_suffix column |
| txtPersonId | Text box | SelCtlSearchTxt?? | Expected for use when a specific person is the focus |
| txtSelRowCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Set by “control update” event procedure to display count of records selected by user’s choices. |

| Control Name | Type | Custom Class | Remarks |
|--------------|------|--------------|---------|
| | | | |

[LB – Seems there may be a need for some additional identifying information, a state of residence, birthdate, related persons, etc]

The user is provided the command options outlined in the section ‘Form – Common “Record Selection Form” Characteristics’ above.

Form – Person Record Selection Subform

Provides a tool to assist in identification of a single person during editing of a “related person” record, a “party membership”, etc. While similar to the “person record selection” form above, this form is designed for use as a control on a “parent” form and does not include command buttons linking to “edit” forms, etc. – the purpose is solely to assist the user to select a single “person record” and make a representation of that selection available to the “parent” form. Because of this, a number of properties and methods are defined that are similar to those defined for “SelCtl...” objects.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|--------------------------------------|---------------------------------|-----------------|---|
| cmdUpdateDisplay | Command button | | Unbound control. Updates display in lstRecDisplay control to reflect user-entered selection criteria. |
| lstRecDisplay | List box, single selection only | | Unbound control. Row source is a subset of rows from person table, possible with additional information from related tables. Provides a compact display of records matching the search criteria entered in other controls on this subform. No default row source – row source is only set in response to user entry of selection criteria – this feature is designed to minimize data transfer from server to client. User selection fills SelClause property of subform. |
| txtFirstNameSearch | Text box | SelCtlSearchTxt | Unbound control. Used by the user to enter text search pattern used to select records for display in associated list box. Wildcards required. Apostrophes removed from search criteria to prevent run-time errors in execution of SQL LIKE operators. |
| txtLastNameSearch | Text box | SelCtlSearchTxt | Unbound control. Used by the user to enter text search pattern used to select records for display in associated list box. Wildcards required. Apostrophes removed from search criteria to prevent run-time errors in execution of SQL LIKE operators. |
| txtSelRowCount [LB – this is still a | Text box, counting number, | | Unbound control. Set by “control update” event procedure to display count of records selected by user’s choices. |

| Control Name | Type | Custom Class | Remarks |
|-----------------------|-----------|--------------|---------|
| "brainstorming" idea] | read only | | |
| | | | |
| | | | |

Module-level variables

mstrSQLBase – portion of SELECT phrase for lstRecDisplay control row source specifying columns to be displayed.

mstrSQLOrder – ORDER clause for lstRecDisplay control row source .

mstrSQLWhere – WHERE clause for lstRecDisplay control row source .

Custom Properties

SelClause – Contains SQL WHERE clause phrase reflecting the user's selection from the lstRecDisplay control.

Custom Methods

SetRecDisplayRowSource – Constructs SQL statement corresponding to user-entered selection criteria and sets the RowSource property of lstRecDisplay to that value.

SetSelClause – Set value of SelClause property to hold appropriate SQL WHERE clause phrase reflecting user selection from lstRecDisplay control. Read-only.

SetSelRowCount – [LB – Still a "maybe", pending a bit more investigation work on implementation mechanisms and performance effects...] Called by SetSelFilter procedure? Sends "row count" query to data server. If non-zero result, sets value of read-only control used to display this information for user. If zero result, displays window informing user, then returns user to last control updated and reverts the last change to that control.

Form – Person Selected Record Display

Provides the user a compact display of information for records matching the criteria entered in the associated "record selection" form.

Controls to be provided include:

| Control Name | Type | Remarks |
|-------------------|---------------------------------|---|
| lstRecDisplay | List box, single selection only | Displays values stored in columns of the person table and closely related tables. Columns displayed include: name_last, name_prefix, name_first, name_middle, name_suffix |
| txtSelDescDisplay | Text box, read-only | Displays the "human-language" version of the filter expression used to select the records displayed in the lstSelectedRecDisplay list box. |

The user will be provided the standard function/command choices outlined under ‘Form – Common “selected record display form” characteristics’ above.

Form – Pledge Edit

Provides the user a display to assist in entry of information characterizing a pledge.

Displayed controls include

| Control Name | Type | Remarks |
|-------------------|---|--|
| lstParty | List box , read-only | Bound to party_id column? Also displaying party name? [LB – Need to decide whether a bound control is really required in this form – currently, I believe it is not needed but may provide helpful clarity to the user.] |
| sfrPartySel | Subform | Provides mechanism for selecting individual party from numerous parties tracked by system. Selection should update txtParty control. |
| sfrRelatedDonRec | Subform, datasheet style? | Displays related donation records. [LB – Will a simple datasheet subform provide adequate guidance for entry of related donation records?] |
| txtPledgeDonCount | Text box, counting number format, read-only | Simply information for the user’s benefit, NOT a value to be entered. |
| txtPledgeId | Text box, counting number format, read-only | Bound to pledge_id column |
| txtRemk | Text box | Bound to remk column. |

The user is provided the command options described in the section ‘Form – Common “Edit form” Characteristics’ above.

Form – Pledge Record Selection

Provides the user a number of unbound controls to be used in selecting a subset of pledge records stored in the system, to facilitate the user’s selection of a record to be edited, or to verify that a new record to be entered does not duplicate an pre-existing record in the system.

Controls to be provided include:

| Control Name | Type | Custom Class | Remarks |
|----------------|-----------------------------|-------------------|---|
| txtPartyName | Text box | SelCtlSearchTxt | Used to search values of party.name |
| txtPledgeDtMax | Text box, short date format | SelCtlIntervalEnd | Used to search for values of pledge_date falling in a specified date interval. Form-based validation to ensure >= txtPledgeDtMin. |

| Control Name | Type | Custom Class | Remarks |
|--|--------------------------------------|-------------------|---|
| txtPledgeDtMin | Text box, short date format | SelCtlIntervalEnd | Used to search for values of pledge_date falling in a specified date interval. Form-based validation to ensure <= txtPledgeDtMax. |
| txtPledgeID | Text box | | Used in cases where an exact pledge is being sought?? [LB – Will this really be used enough to make it worth providing?] |
| txtSelRowCount [LB – this is still a “brainstorming” idea] | Text box, counting number, read only | | Set by “control update” event procedure to display count of records selected by user’s choices. |

[LB – Is there a need for additional record selection criteria? Donation date range? Donation amount interval? Other?]

The user is provided the command options outlined in the section ‘Form – Common “Record Selection Form” Characteristics’ above.

Form – Pledge Selected Record Display

Provides the user a compact display of information for records matching the criteria entered in the associated “record selection” form.

Controls to be provided include:

| Control Name | Type | Remarks |
|-------------------|---------------------------------|---|
| lstRecDisplay | List box, single selection only | Displays values stored in columns of the ?? table and closely related tables. Columns displayed include: pledge_id, pledge_date, party.name (linked through relationship to party), remk (1 st 20 characters). |
| txtSelDescDisplay | Text box, read-only | Displays the “human-language” version of the filter expression used to select the records displayed in the lstSelectedRecDisplay list box. |
| | | |

The user will be provided the standard function/command choices outlined under ‘Form – Common “selected record display form” characteristics’ above.

Form – Report Switchboard

Provides the user a compact display of information for records matching the criteria entered in the associated “record selection” form.

Controls to be provided include:

| Control Name | Type | Remarks |
|--------------|----------------|--|
| cmdRptA | Command button | Opens “record selection form” appropriate to “Report A”, fills in selection criteria appropriately, and sets filter expression accordingly. Passes “output format” setting to be passed to “selected records display” form from “record selection form”. |
| | | |
| | | |

[LB – The number of command buttons will correspond to the number of standard reports established for the application. There may be a need for more than one “command button” for each report if the same report may be generated for multiple data areas. For instance, mailing lists might be generated from donations, parties, or contacts – the user must be offered the option of selecting which subject area is to be used for composing the report.]

Form – Staff Member Edit

Provides the user a display to assist in entry of information characterizing a staff member record.

[LB – I think a simple datasheet form may be adequate for this function.]

Displayed controls include

| Control Name | Type | Remarks |
|-------------------|---|--|
| chkCurrentStaffYN | Check box?? Option group?? | Bound to current_staff_yn . Default value “yes”. |
| txtAbbr | Text box | Bound to abbr . Table-level validation to require unique abbreviations – not clear whether form-level validation is worth enough to warrant effort. |
| txtNameFirst | Text box | Bound to name first |
| txtNameLast | Text box | Bound to name last |
| txtStaffMbrId | Text box, counting number format, read-only | Bound to staff_mbr_id |
| | | |

The user is provided the command options described in the section ‘Form – Common “main record Edit form” Characteristics’ above. [LB – If the decision to use a datasheet is adopted, this form will not follow all the characteristics of the “main record edit form”.]

Code Modules

This section contains descriptions of all public code modules to be provided in the new system. These modules will provide functionality used by multiple forms/code modules in the final application.

Module - AutoExec

This module will:

1. Download lists of all static data used to populate combo and list box controls, etc. This is done to prevent frequent queries from client to data server during form use, etc. for purposes of enhancing client/server performance.
2. Initializes global function and procedures.

Procedure “DownloadServerData” – This procedure downloads data used repeatedly in lookups, control row sources, etc. from server to the client application at application startup. The purpose is to improve performance by downloading static, unchanging data that is used repeatedly by the client application a single time, rather than passing data from server to client each time this data is requested. Data targeted for this type of transfer will typically be code definitions. [LB – Some provision for “refreshing” changed data from server to clients may be needed in cases where an administrator changes definitions while users are using the application. Otherwise, administrators may be required to restrict these changes to periods when the system is not being used.]

Module – Global Functions/Procedures

Function “GetRptSelDesc” ?? - Used to retrieve description of selection clause and insert as report subtitle or similar when reports are generated. [LB – Can this be substituted by appropriate assignment of control values directly from a “selected records display” form?]

Function “GetRowCount” – Is passed a base SQL “select” statement and a corresponding filter location, queries server and returns the row count. This type of functionality may be desirable in many “record selection” forms in the application – it may consequently be desirable to use a global function rather than a “form module-level” function.

Procedure “AddToCommaDelimList” ?? – Is passed a reference to a base string, and a string value to append. Appropriately adds the append string with use of commas, spaces, etc to create a comma delimited string suitable for use in an SQL “IN (*string*)” clause. [LB – This type of function is used in several class modules planned for this application. Not clear at present whether duplication and inclusion of this functionality within each class module is preferable to creation of a shared, global procedure. The duplication provides a very tight modularization of class methods that may be desirable.]

Procedure “BuildFilter” – Is passed a reference to a base string, and a string value to append, and an indication of the interaction of the new clause in the combined where clause (“AND”, “AND NOT”). Appropriately adds the append string to the base string. [LB – Some potential overlap with functionality provided in some procedures of class modules planned for this application.]

Module – “SelCtlGrp” Class Definition

This module will define the class used to define methods, events, and properties that will streamline the use of multi-select list box controls for “group membership” on “record selection forms” used repeatedly in the application. Methods required for this purpose differ somewhat from those used with the “SelCtlMList” class because of the use of “start date”, “end date” and “effective date” parameters. For this reason, a different class is used.

Assumes that column 0 of list box contains the numeric id value(s) to be used in filter expression. Column 1 is assumed to contain a descriptive abbreviation for that id value

Properties include:

DtCtl – Indicates text box containing date used to determine membership in selected groups.

LstCtl – Indicates multi-select list control to be used in class instance.

IdInteract – Takes values indicating whether application of multiple selections should use “AND” logic or “OR” logic.

SelClause – Contains SQL clause implementing the currently entered values from the control

SelDesc – Provides a description of the selection criteria indicated in the SelClause property designed for inclusion in report subtitles, etc. Column names used in SelClause are replaced with more labels that are a bit more friendly to end users.

SelType – Takes values reflecting “Mbr” or “NotMbr” use of the user-selected groups in the list box.

TrgtEndDtCol – The name of the “membership period end date” column in the “group membership period” table.

TrgtIDCol – Holds name of the group ID column in the “group membership period” table.

TrgtStDtCol – The name of the “start date” column in the “group membership period” table.

TrgtTable – Holds name of the table holding “group membership period” data.

Methods include:

BuildCommaDelimList – Used internally to construct lists of selected groups.

OneIDTestClause – Used internally to construct a test clause for a single selected group.

ReqsOK – Used internally to check that all required properties have values specified before the SetSelClause method is executed.

SetSelClause – Constructs appropriate value for “SelClause”. The method executes only if all properties have defined values.

SetSelDesc – Constructs appropriate value for “SelClause”. Executed each time the SetSelClause method is invoked.

Module – “SelCtlIntervalEnd” Class Definition

This module will define the class used to define methods, events and properties that will streamline the use of text box controls to specify dates used to define time windows for record selection. Methods required for construction of filter clauses employing date values differ enough from those employing numbers to warrant a separate class.

[LB – Could be generalized to any text box provision of a value, whether date, number, character, and to allow all operators (“<”, “=”, “>”, “<=”, “>=”). Probably a change of class name would be appropriate in that case...]

Properties include:

Ctl – Indicates text box control containing value to be used in class instance.

IntervalType – Indicates whether interval being defined is date/number.

SelClause – Contains SQL clause implementing the currently entered values from the control

SelDesc – Contains “natural language” version of filter clause implementing the currently entered values from the control

SelType – Takes enumerated values reflecting “<=” or “>=” use of the values provided by the control.

TrgtCol – The name of the column targeted by the selection

TrgtTable – The name of the table holding the column targeted by the selection.

Methods include:

SetSelClause – Constructs appropriate value for “SelClause” property. The method executes only if all properties have defined values.

SetSelDesc – Constructs appropriate “human language” description of “SelClause”. Executed each time the SetSelClause method is invoked.

Module – “SelCtlMultList” Class Definition

This module will define the class used to define methods, events, and properties that will streamline the use of multi-select list box controls for basic “category identification” on “record selection forms” used repeatedly in the application.

[LB – I think this could be easily generalized to allow use with either single- or multi-select list boxes. Generalization to also include combo box controls may also be desirable.]

Assumes that column 0 is contains the values to be used in filter statements.

Properties include:

Ctl – Indicates multi-select list control to be used in class instance.

SelClause – Contains SQL clause implementing the currently entered values from the control

SelDesc – Contains natural language version of SelClause value.

SelInteraction – Takes values indicating whether combined application of multiple selections should use “AND” logic or “OR” logic.

SelType – Takes values reflecting “IN” or “NOT IN” use of the user selection.

TrgtCol – The name of the column targeted by the selection

TrgtTable – Holds name of the table holding the column targeted by the selection.

ValType – Indicates whether value to be used is number/character (alters form of filter statement)

Methods include:

SetSelClause – Constructs appropriate value for “SelClause” property. The method executes only if all required properties have defined values.

SetSelDesc - Constructs appropriate value for “SelDesc” property. Executed whenever SetSelClause method is run.

Module – “SelCtlTxt” Class Definition

This module will define the class used to define methods, events and properties that will streamline the use of text box controls to specify numbers used to define single values for record selection.

Properties include:

Ctl – Indicates control to be used in class instance.

SelClause – Contains SQL clause implementing the currently entered values from the control

SelDesc – Human-language description of SelClause property value.

SelType – Default “=”, may be no need for this property at all in this class. [LB – Alternatively, if SelCtlIntervalEnd is generalized to allow different SelType values, then this class may be unnecessary.]

TrgtCol – The name of the column targeted by the selection

TrgtTable – Holds name of the table holding the column targeted by the selection.

ValType - Indicates whether value is character/number.

Methods include:

SetSelClause – Constructs appropriate value for “SelClause” property. The method executes only if all properties have defined values.

SetSelDesc - Constructs appropriate “human language” description of “SelClause”. Executed each time the SetSelClause method is invoked.

Module – “SelCtlTxtPattern” Class Definition

[LB – Not finalized that this class is really needed...]

This module will define the class used to define methods, events and properties that will streamline the use of text box controls to specify text search patterns used for record selection.

This module may also need features to appropriately deal with “special characters” that might be part of a text string (apostrophe, quote, etc.).

Properties include:

Ctl – Indicates control to be used in class instance.

SelClause – Contains SQL clause implementing the currently entered values from the control

SelDesc – Contains human-language version of SQL filter clause reflecting this control’s setting.

SelType – Takes values reflecting “LIKE” or “NOT LIKE” use of the value provided by the control.

TrgtCol – The name of the column targeted by the selection

TrgtTable – Holds name of the table holding the column targeted by the selection.

Methods include:

SetSelClause – Constructs appropriate value for “SelClause” property. The method executes only if all properties have defined values.

SetSelDesc - Constructs appropriate “human language” description of “SelClause”. Executed each time the SetSelClause method is invoked.

Module - ?

Reports

This section contains descriptions of all reports designed for use in the new system. Many of these will be used by several other forms/code modules in the system.

All reports will be defined as part of the “client” portion of the application.

All “data source” properties will be set using SQL statements assigned to that property. Stored queries that will be visible to the user within the MS-Access “query” collection will be avoided.

***Report – Contacts**

This report displays a list of contact records.

Applies a filter clause supplied by the calling module.

Information displayed includes:

- Donor relationship name

- Staff member name

- Contact subject(s)

- Notes from the contact

- Contact date

Report – Easement Project Landowner

Displays information about parties involved in easement projects in the role of “landowner”.

Displayed information includes:

- Easement project

- Property name

- Easement project manager

- Name of primary staff contact

- Party name

- Name of primary contact for the party

Name of spouse for the primary contact

For all locations associated with the “party”:

Location type

Street address

City

State/Province

Country

Postal Code

Phone numbers associated with the location

Phone numbers associated directly to the “party” rather than to specific locations

Name of primary staff contact

All “party category” codes assigned to the “party”

All “groups” to which the “party” belongs at the time the report is generated. [LB – Since we expect groups to replace many of the current “status codes”, it seemed like these should be included.]

Property manager/caretaker [LB – I’m not sure we’ve explicitly included a spot for this information in the current data model. I suppose the caretaker is a “party” – someone who’s address, email, etc you’ll want to track. And that a “easement_proj_party” record should be entered linking that “party” to the easement project and indicating the role of “caretaker” or “property manager”.]

The displayed information is arranged by easement project

Report – Donation/Budget Comparison

***Report – Group Members**

This report displays a list of parties belonging to a specified group as of a user-provided “effective date”. I proposed this mostly an on-screen group membership review tool for database users.

Information displayed includes:

Relationship ID

Relationship name

For primary relationship contact:

First name

Last name

Membership start date?

Membership end date?

***Report – Land Steward Easements**

Displays a list of all easements assigned to each land steward..

Information displayed includes:

Easement project ID

Land steward name

Name of primary staff contact for the landowner involved in the easement [LB – Have I got this right? Or is it the “easement project manager” that you’d be more likely to want?]

Locations

***Report - Locations**

Displays a list of all locations.

Information displayed includes:

For each location:

Location ID

Attributes of location (address, city, etc.)

The name of the primay contact for each “party” associated with the location

Report – Mail Merge File

DOS file containing information required to perform a basic letter merge for a select group of individuals. Set up specifically to feed information to MS-Word letter merge operations. The information displayed characterizes a group of relationship/persons matching a set of selection criteria specified by the user:

Information used will include:

- Relationship name

- For primary relationship contact:

 - First Name or Nickname?

 - Last Name

 - Name prefix

- For primary location:

 - Street Address

 - City

 - State/Province

 - Postal Code

***Report – Mailing Label**

Generates mailing labels for party records.

Uses Avery 5162 address labels for printing.

Applies record filter constructed by calling application. Filter should always include provisions to use only the primary contact location for the party.

Includes the following attributes:

- Party name

- Street address

- City

- State/Province

Postal code

Country

Desirable to automatically insert bar code representing postal code on labels.

[LB – Is there a specific ordering of selected records that is desirable?]

***Report – Location File**

This will be a DOS file that contains the information needed for a mailing service to perform a mailing assembly operation. The file will be a DOS text file with tab-separated fields.

Information used will include:

Relationship ID

Relationship name

Street Address (for primary contact location)

City (for primary contact location)

State/Province (for primary contact location)

Postal Code (for primary contact location)

Country (for primary contact location)

***Report – Address File**

This report provides information that might be exported from the database for a staff member's inclusion in an address book or similar.

Information used will include:

Party name

Name of primary contact person for the party, including:

First Name or Nickname of primary contact person for the party

Last Name of primary

Name prefix

Street Address

City

State/Province

Postal Code

Primary email address

Primary phone number

***Report – Donation Detail**

This report lists information about each donation. The donations are ordered by party and then by donation date.

Information displayed includes:

Donor party name

For each donation matching the selection criteria:

Donation date

Amount

Pledge (yes/no)

Pledge amount

Proxy donation (yes/no)

Proxy donation original donor relationship name

Fund contributed to

Sum of “amount” values

Sum of “pledge amount” values

***Report – Party Donation Summary**

(Very similar to the Donation Detail report, but with concise summaries displayed for each individual rather than detailed displays of each donation.)

This report lists summary information for all donations made by parties. The displayed information is ordered by party name.

Information displayed includes:

Donor relationship name

For each fund contributed to:

Fund(s) to which the individual/relationship donated during the time period

Total amount contributed to the fund

Report - ?