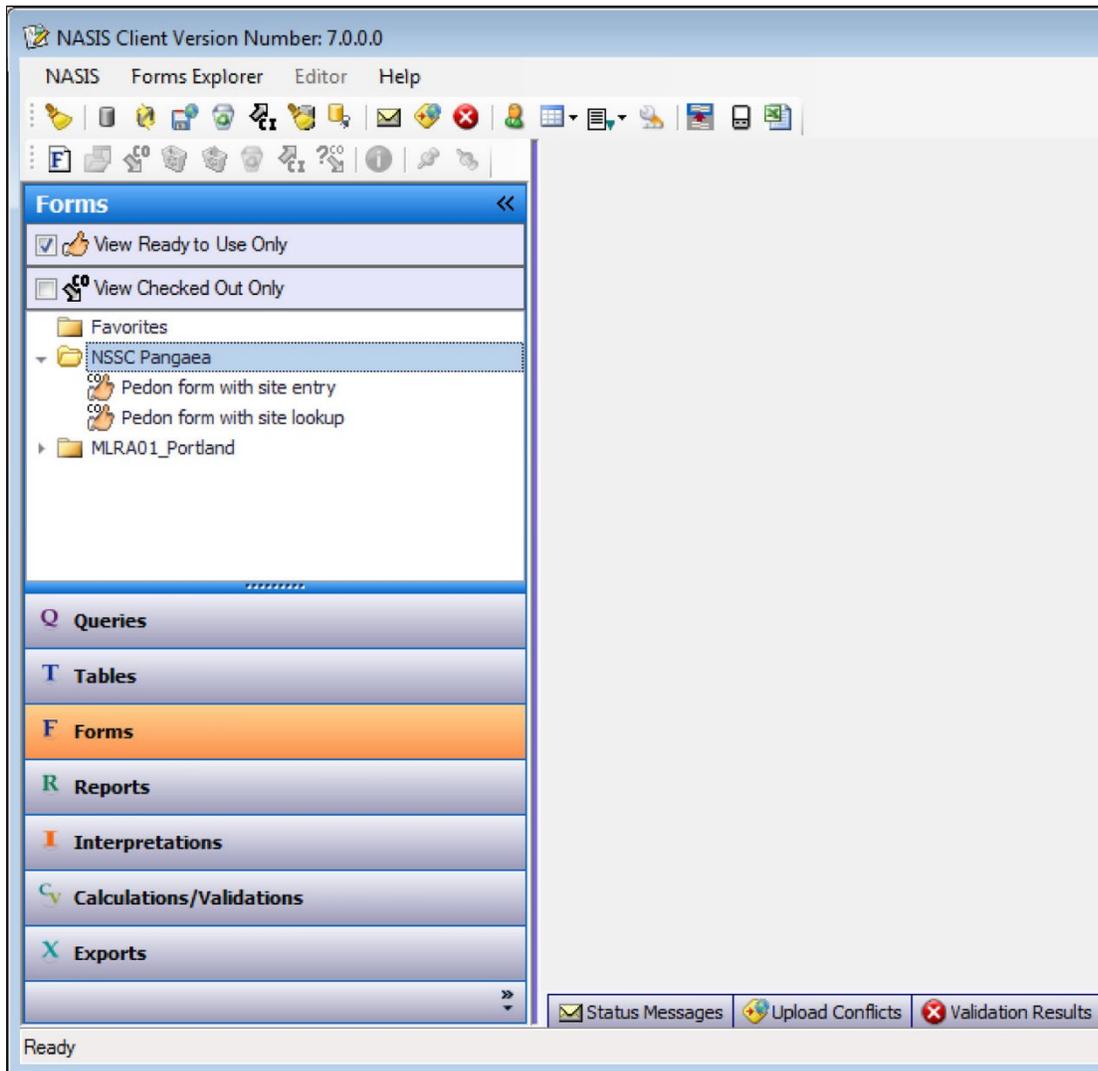


Chapter 26: NASIS Forms Module

What Are NASIS Forms?

The NASIS Forms module provides a new way to view and edit data in NASIS. All of the familiar NASIS procedures apply to form-based editing, including downloading data, creating a selected set, checking out the data, editing it, and uploading changes. With forms, you can design a custom screen layout as an alternative to using the traditional table layout.

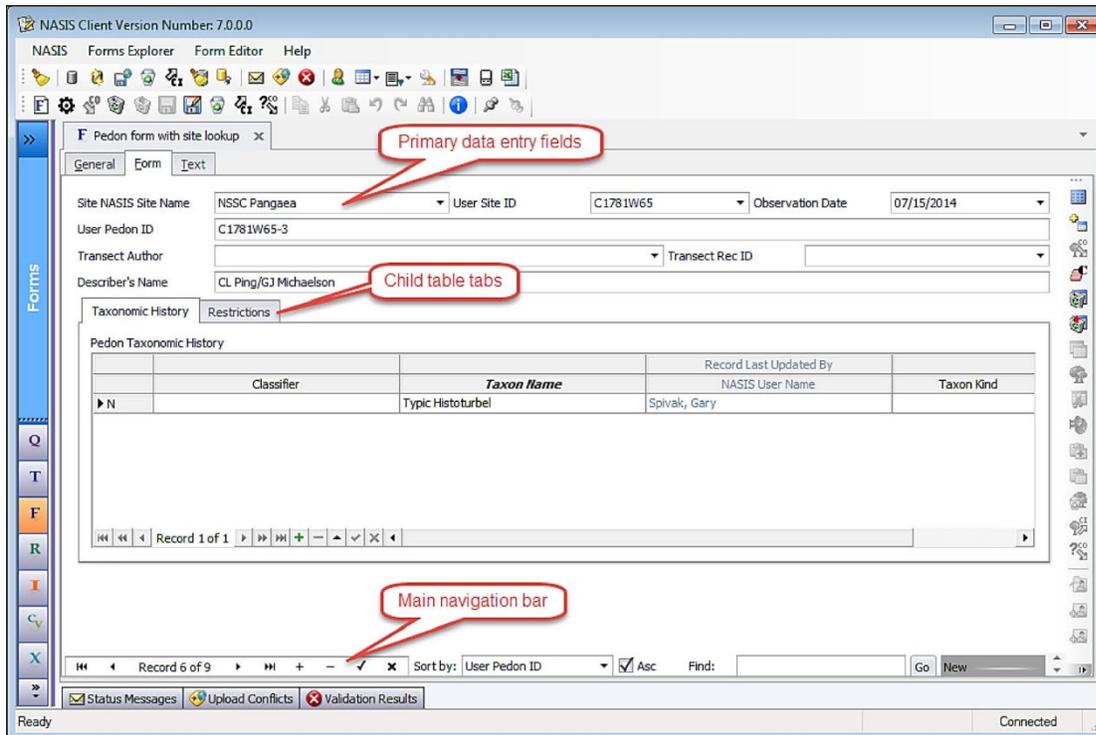
A new **Forms Explorer** module allows you to select a form by region and name.



The Forms Explorer module functions just like Queries or other existing modules. Anyone can create a new form. After it is uploaded, others can use the form; however, only those in the group that owns the form may check it out and modify it.

Entering Data using NASIS Forms

The following screenshot shows what a form looks like when opened from the Forms Explorer.



This form is designed around the Pedon table, which is called the *base table* for the form. Two types of data entry fields are shown. Primary data entry fields have a label to the left or right and are part of the base table or part of one of its parent tables. Child tables are displayed as familiar NASIS tables. NASIS tables can be arranged in tabs (as in this example) or arranged as separate side-by-side tables. Each child table has its own navigation bar to control the records in that table.

The main navigation bar, at the bottom of the screen, is used to move from one base table record to another in the selected set. In this example, the main navigation bar shows that the selected set contains nine Pedon records. Clicking the plus icon ("+") in the navigation bar displays a blank form used to add a new Pedon record. The minus icon ("-") deletes a record (provided that it is checked out). The record status appears at the far right of the navigation bar: **New**, **Checked Out**, **Modified**, **Deleted**, and so on. The record status should not be confused with the form status, which is displayed with an icon on the main navigation bar.

To the immediate right of the navigation bar is a sort selector (the **Sort by** drop-down list). Records displayed in the form can be sorted on any of the primary data entry fields. If, in addition, the checkbox **Asc** is selected, the sort is in ascending order; if **Asc** is not selected, the sort is descending. A **Find** box lets you search the selected set for a match in any of the primary data entry fields.

Note that not all of the fields in the Pedon table, or in any of the other tables, are shown on this form. The designer of each form controls which fields will appear on that form. If you need to enter data in a field that is not included on the form, use a different form or use the table editor.

NASIS User Guide

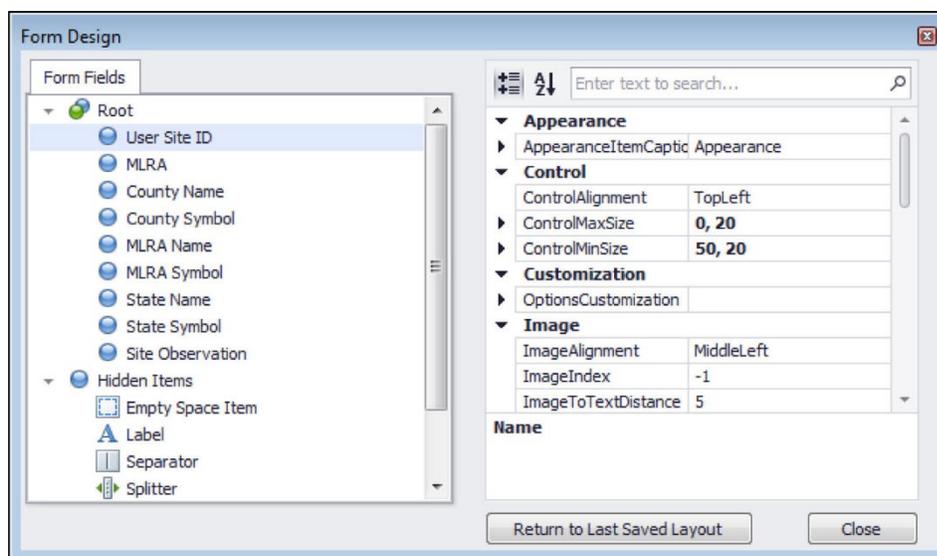
On this form, several of the primary fields contain a drop-down arrow that presents a list of choices from which you must select one item. When entering information for a new Pedon, you type the **User Pedon ID**, but you must select the **User Site ID** and **Observation Date** from a list of Site and Site Observation records stored in the local database. Alternatively, the form designer could set it up so that these fields do not have choice lists and you have to type the **User Pedon ID**, **User Site ID**, and **Observation Date** into the form. In that case it would create new records in all three tables at once.

Customizing the NASIS Form

Even if your user account does not have permissions to check out a form, you can customize some aspects of the form layout as described in the following subsections. These customizations for your own use are saved on your computer but are not shared with other NASIS users.

Modify the Form Design

On the **Form Editor** menu, selecting a new option called **Design Form**, represented by a gear icon (⚙️), temporarily disables data entry on the form and instead displays the following **Form Design** dialog box.



On the left, the **Form Fields** list contains two groups: **Root** lists all items currently displayed on the form, and **Hidden Items** are items that you can add to the form. You can click an item in either list and drag it to the other list; dragging from **Root** to **Hidden Items** hides the item, and dragging from **Hidden Items** to **Root** adds an item to the form. You can also drag an item from **Hidden Items** directly onto the form.

On the right, properties of the item currently selected in the **Root** list are displayed (the right column is blank when a **Hidden Item** is selected). You can change attributes such as the color, font, and text for one or more field captions by expanding the headings **Text** or **AppearanceItemCaption**

NASIS User Guide

The **Hidden Items** list contains several standard items that you can place on the form to improve its appearance or usability. More than one of each standard item can be added to the form. When added, the item appears in the **Root** list. If you add multiple instances of the same item, it is helpful to change each item's text so that you can distinguish each from the others.

The items listed under **Hidden Items** are:

- **Empty Space Item**

A blank rectangular area that you can place between, above, or below fields to open up the layout. You can adjust the height of an empty space area by dragging its top line. You can set the background color of the area after it has been added to the form.

- **Label**

A rectangular area that contains fixed text (specified by you) that you can place anywhere on the form. For this label, you can adjust properties such as text size, spacing, font, and color.

- **Separator**

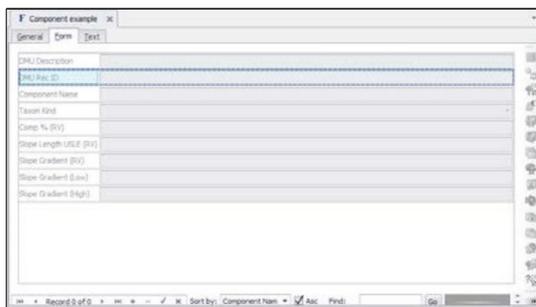
A fixed horizontal or vertical line that you can place between fields on a form. You can use its **Spacing** property to create space around the line.

- **Splitter**

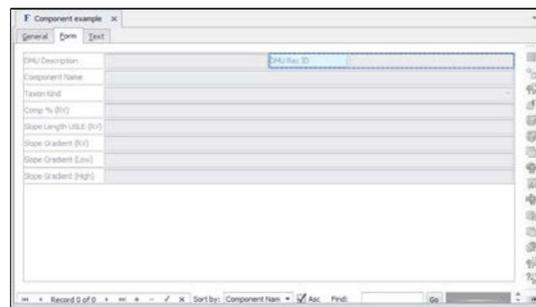
Similar to a separator but adjustable even when the form is not in design mode. Instead of appearing as a fixed line, a **Splitter** appears as a short line of dots. The user of the form can click the dots and slide the splitter's position. For example, a vertical splitter between two fields or tables allows the user to make them wider or narrower. A horizontal splitter above an empty space area allows the user to change the height of the empty space.

Modify the Form Layout

Form Design mode also lets you reposition items on the form. Clicking an item on the form displays a dashed line around the item; now, you can drag the selected item across the form. A black line or rectangle will appear at any location on the form where the item can be dropped. Adjacent items on the form are then moved to accommodate the repositioned item.



Default field layout



Field repositioned

When two or more items are displayed in a row across a form (as in the *Field repositioned* screenshot above), you can click on the line between two fields and slide the line left or right to change the size of the fields. Typically, you cannot change the height of fields except for empty space areas.

NASIS User Guide

The size of the fields on a form is normally set automatically so that all the fields fit onto the screen, and size properties cannot be modified in the Form Design dialog box. However, by changing the **Size Constraints Type** to **Custom**, you can set the minimum and maximum sizes of individual fields. Such changes might affect neighboring fields. By setting minimum sizes for several adjacent fields you can create a form layout that exceeds the size of the screen, which will trigger the appearance of scroll bars on the form.

When a table appears on a form, you can select the table and drag it to a new position. It is often useful to place two or more tables side by side so that readers can easily compare data in the tables. You cannot change the arrangement of columns in a table while in Form Design mode, but you can rearrange columns while editing the table. This feature is also available in the NASIS table editor. You can move or resize columns, and your changes are saved and available the next time that you use this form. Be aware that modifying a form in Form Design mode resets the columns of all tables on this form to their initial position and width.

The screenshot shows a form titled "Component example" with tabs for "General", "Form", and "Text". The "Form" tab is active. The form contains several input fields and two tables.

Input fields include:

- DMU Description: 00 IBC
- DMU Rec ID: 63598
- Component Name: BATES
- Taxon Kind: series
- Comp % (Low): [empty]
- Comp % (RV): 85
- Comp % (High): [empty]
- Slope Gradient (Low): 2.0
- Slope Gradient (RV): 5.0
- Slope Gradient (High): 7.0

Table 1: Horizon

Designation	S	Top Depth			Bott
		Low	RV Δ	High	
H1			0		
H2			15		
Cr	M		58		

Table 2: Component Restrictions

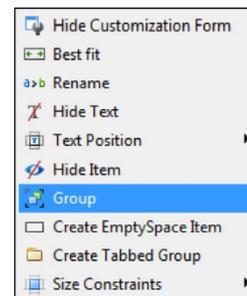
Top ...	Bott...	Kind
RV Δ	RV Δ	
58		bedrock, paralithic

Form that displays two tables side by side

Group Fields on the Form

On a form, when you select one or more fields or tables, you can right-click in the selected area to display a pop-up menu with layout options. Some of these layout options are duplicates of settings that can be modified by using the Form Design dialog box, but this pop-up menu provides additional options for creating groups and/or tabbed groups. These are powerful options for organizing a form to use space efficiently.

The **Group** option creates a box around the selected field(s) with a generic label such as **item4**. Right-click again in the group area to display options to modify the label text and position, or to hide the label. After you create a new group, the group name will also appear in the **Root** list of the Form Design dialog box, and the new group's enclosed fields will be shown under it. You can also set group properties in this dialog box.



NASIS User Guide

The **Create Tabbed Group** option is similar to the **Group** option, but it creates a tab for the selected field(s) and encloses the set of tabbed groups in a box. Each tab displays a generic name that is visible by default; the surrounding box is assigned another name that is not visible by default. This results in two new levels in the **Root** list, where the properties such as text and color can be modified.

The usefulness of a tabbed group is apparent when you create more than one tab. After creating the initial tabbed group, create another group as before: select some fields or tables that are not in the first group, right-click and choose **Group**. Specify a name for the group. Then click in the heading area of the new group and drag it to a position immediately to the right of the tab for the first tabbed group. A small rectangle indicates the position for the second group. When dropped, the second group becomes a new (second) tab in the tabbed group.

The screenshot shows a web-based form editor window titled 'Component example *'. It has three tabs: 'General', 'Form', and 'Text'. The 'Form' tab is active, displaying a form with several input fields and a table. The fields include 'DMU Description' (001BC), 'DMU Rec ID' (63598), 'Component Name' (BATES), 'Taxon Kind' (series), 'Comp % (Low)', 'Comp % (RV)' (85), 'Comp % (High)', 'Slope Gradient (Low)' (2.0), 'Slope Gradient (RV)' (5.0), and 'Slope Gradient (High)' (7.0). Below these fields is a tabbed group with two tabs: 'Horizon' (selected) and 'Restrictions'. The 'Horizon' tab contains a table with columns for 'Designation', 'S', 'Top Depth' (Low, RV, High), and 'Bottom Depth' (Low, RV, High). The table has two rows: H1 and H2. The 'Restrictions' tab is currently empty. The bottom of the window shows a status bar with 'Record 1 of 4', a 'Sort by: Component Nam' dropdown, a checked 'Asc' checkbox, a 'Find:' field, a 'Go' button, and a 'Not Checked Out' indicator.

Designation	S	Top Depth			Bottom Depth		
		Low	RV	High	Low	RV	High
H1			0			15	
H2			15			58	

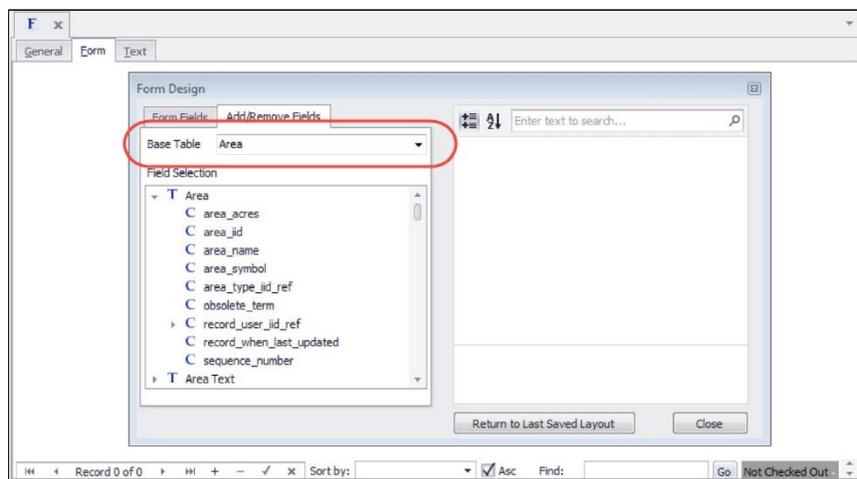
Form with two tables that comprise a tabbed group

Creating a New NASIS Form

On the **Forms Explorer** menu or toolbar of the NASIS Forms module, select **New Form**. This opens the **General** tab of the form editor, where you must enter a name for the form and where you can, optionally, add a description and/or change the ownership of the form. A **Text** tab is also available for recording notes, such as documentation of changes to the form.

Next, select the **Form** tab, which displays a blank page. Select **Design Form** from the editor menu (or right-click in the blank area of the form) to display the **Form Design** dialog box. This dialog box provides features described earlier for customizing an existing form and additional features that are available only when creating a new form (or editing a checked-out form).

NASIS User Guide



On the left, a new tab called **Add/Remove Fields** appears. Click **Add/Remove Fields**, and then select a **Base Table** from the drop-down list. The base table is the first table read from the database when using the form; everything else on the form must be directly related to the base table. When editing an existing form, you cannot change the base table.

When you select a base table, the **Field Selection** list displays all NASIS tables that are directly related to the base table, and the base table itself expands to show its columns. The **T** icon indicates a table and the **C** icon indicates a column. When you right-click or double-click a column name, a pop-up presents the option to **Add to Form** (or **Remove from Form** if the column is already on the form). Click **Add to Form** to create a field at the bottom of the form containing the column's label and a data entry area appropriate for the column's data type. If other fields are on the form, you can drag the new field to a different position above or beside an existing field. Add as many columns from the base table as you need.

In the columns listed under **Field Selection**, some display an *expand icon*. Click an expand icon to display additional columns called **Lookup Columns** that are marked with an **L** icon. The parent column of a set of **Lookup Columns** is a *foreign key* that references some other NASIS table. If the foreign key column itself is added to the form, the form displays only the internal record number; alternatively, if lookup columns are added to the form, each column displays a choice list that lets you select the record to which you wish to link.

For example, if Pedon is the base table, the column **site_observation_iid_ref** displays an expand icon, which expands to present a list of lookup columns that can be added to a form. Each lookup column added to the form will be displayed on the form as a choice list for selecting a **Site Observation** to which to link the Pedon.

NASIS User Guide

Form Fields Add/Remove Fields

Base Table Pedon

Field Selection

- C sas_pipe_length_unfilled
- ▼ C site_observation_iid_ref
 - L nasis_site_name
 - L observation_date
 - L observation_date_kind
 - L site_iid
 - L site_observation_iid
 - L user_site_iid
- C soil_replicate_number
- C taxon_kind
- C taxon_name

Foreign key and lookup columns

Site NASIS Site Name	User Site ID	Site Rec ID	Observation Date
NSSC Pangaea	C1781W65	1119086	2014-07-15 00:00:00Z
NSSC Pangaea	C1781W65	1119086	2014-07-15 00:00:00Z
NSSC Pangaea	C1781W65	1119086	2014-07-15 00:00:00Z
NSSC Pangaea	C1781W65	1119086	2014-07-15 00:00:00Z
NSSC Pangaea	C1781W65	1119086	2014-07-15 00:00:00Z

Lookup column on a form with choice list

Adding Tables to the NASIS Form

After you add base table fields to a form, you can add other tables to the form. Right-click a table name to display the options **Add Child Table to Form** and **Merge Tables**. If these options do *not* appear, this indicates that the table cannot be added yet to the form. There has to be a pathway of directly linked tables on the form from the base table to the table being added. For example, if Pedon is the base table, you must add the Pedon Horizon table before you add any of the tables that are a child of Pedon Horizon.

After you add a table, double-click the table name (or click the expand icon next to the name) to display the list of columns. You can add these columns to the form as described earlier.

In a few tables there are columns that do not appear in the list because they contain a type of data that is not currently supported in forms. For example, in the Pedon Horizon VNIR Scan table there is a VNIR File Name column that cannot be added to a form. In the Table view, this column is used to import a file, but that does not work in forms. The VNIR File Name is a required field in this table, which means that you cannot enter new rows in this table via a form. There are also columns in some tables that contain URLs, and they do not appear in the list of columns to add to a form.

There are two methods to add a table to a form, **Child Table** and **Merge**, and they operate differently. It is important to understand how each works:

- A child table is displayed in a format similar to a table in the NASIS Table Editor.
- Merge Tables combines the columns of a new table with the columns of a table already on the form; the result is that the two tables appear as one table.

NASIS User Guide

Note: When editing data in a table a special toolbar appears to the right of the table. The controls on this toolbar function as you might expect, but only on forms that have been saved and opened since they were last saved. If you create a new form, the controls on the toolbar do nothing.

Add a Child Table

When you add a child table to a form, it first appears as a narrow band across the bottom of the form. It has a heading, which appears on the properties panel as the **Text** property; you can change the heading text if you wish. You can also change the heading text location and alignment, or you can make the heading invisible. In addition, you can use the **AppearanceItemCaption** properties to change font, color, and border.

The newly added table is empty, so the next step is to expand the table item in the **Add/Remove Fields** list to see the table's columns. As you add columns to the form, the columns are positioned left to right in the table. Columns look like columns in the NASIS Table Editor. Finally, you can drag the table's heading area to place the table where you want it on the form.

Merge Two or More Tables

When you merge two or more tables onto a form, the underlying database records are joined together and displayed as if they were a longer (single) record. When a user enters new data on a merged record, new records are created in each of the underlying tables.

Consider each of the following cases:

- 1. The merged table is a parent of the base table.** A special notation is used for parents of the base table in the **Add/Remove Fields** list: **parent table of child table**. For example, if Pedon is the base table, you will find **Site Observation of Pedon** and **Site of Site Observation** in the list. This convention is used because a given table could be a parent of more than one table, so the "of" is used to clarify the relationship. For these tables, the option **Add Child Table to Form** is *not* available, because these are *not* child tables.
 - When merging a parent table with the base table, a unique parent record is matched with each base table record, and the combination appears on the form if both records are in the selected set. In some cases, this can cause confusion, such as when a Pedon is merged with a Site Observation (and possibly also with a Site). If a Pedon record is in the selected set but its associated Site is not in the selected set (or perhaps not in the local database), that Pedon does not appear on the form.
 - A related issue has to do with permissions. In this merge example, if a Pedon is checked out but its Site record is not checked out, the merged record cannot be edited and its status appears as **Not Checked Out**.
- 2. The merged table is a child of the base table.** With any parent-child relationship in the database, the possibility exists that zero, one, or more than one child record exists for a given parent record. The form's behavior varies accordingly:
 - **No child records.** When no child record is found for a given parent record, fields derived from the child record are blank. If data are entered in any of those fields, a new child record is created.
 - **One child record.** When exactly one child record is found for a given parent record,

NASIS User Guide

the combined record is edited in the same manner as described earlier for merging a parent table.

NASIS User Guide

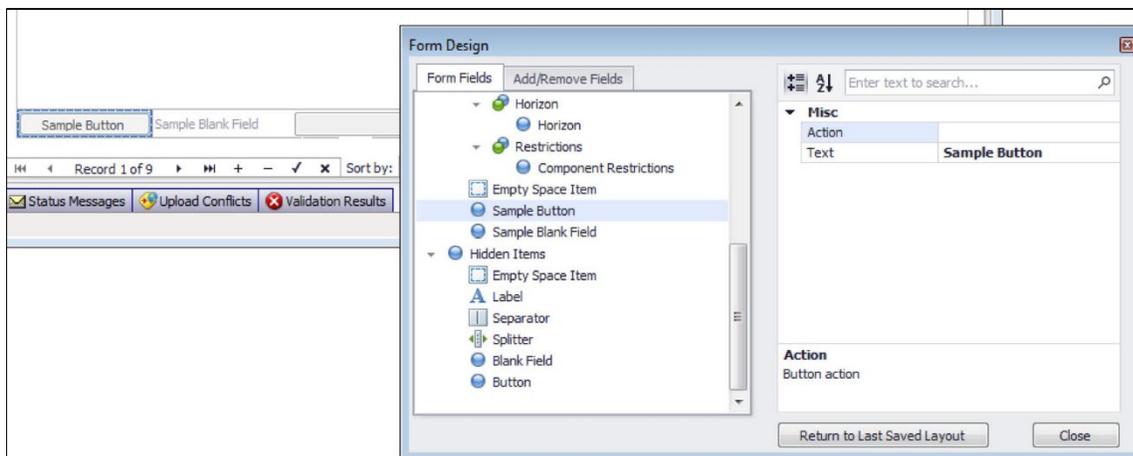
- **Multiple child records.** When multiple child records are found for a given parent record, the merge results in multiple combined records. This gives the appearance that there are more base table records in the selected set than actually exist. However, these are not really duplicate base table records. If a change is made to a field derived from the base table, that change will appear in each of the combined records as you navigate through the records in the form.
3. **The merged table is a child of a child table.** After a table is added to the form as a child table, it is possible to merge its child tables. The operation is similar to merging a child to the base table, but the combined records are displayed in table format. When a one-to-many relationship exists, the number of rows in the table increases to include all the combinations.

Adding Blank Fields, Buttons, and/or Scripts

When you create a form, two additional items appear in the **Hidden Items** list, **Blank Field** and **Button**. Like other items on the **Hidden Items** list, you can drag one or more instances of either of these onto the form. Initially, a dragged-and-dropped item is assigned a default name, which you should change so that the item can be uniquely identified.

A **Blank Field** is a labeled text box that does not accept input and is not tied to any database column. It can be used for temporary data entry or to display output from a button action. The blank field has a property called **Multiline**, which is set to **False** by default. If **Multiline** is set to **True**, the blank field can display more than a single line of text and its height on the form can be adjusted by dragging the bottom edge.

A **Button** has two properties that can be assigned in the property panel *after* the button is placed on the form: **Text** and **Action**. **Text** is displayed on the button, whereas **Action** is a script that runs when the user clicks the button. The typical result of a button action (script) is to modify the contents of one or more fields (including blank fields) on the form.



Button and Blank Field added to a form

The **Action** field on the properties panel is small and not convenient for entering the script. Instead, right-click on the button's name in the Form Fields list and select **Edit Script**. This displays an editing panel for the script.

NASIS User Guide

The **Action** script is a subset of the **NASIS CVIR** script used for reports and calculations. This script can include queries (EXEC SQL statements) and DEFINE statements that operate on data. The complete syntax for CVIR scripts is found in the CVIR Language manual at http://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053305.pdf.

The unique features of a form action script are:

- Data on the form can be accessed in the script by using the field label as a variable. Any label that includes spaces or special characters has to be enclosed in square brackets, such as [Comp % (RV)].
- When a field on the form is a column in a child table, the associated variable in the script can have multiple values, and it can be operated on like other multi-valued CVIR variables.
- The SET statement is used to place calculated values onto the form. Again, the field label is used to identify the field on the form where the value will be placed. Either normal database fields or blank fields can receive data this way.

For example:

```
SET [Sample Blank Field] FROM some_variable.
```

- The WHEN statement can be used to display a message in the **Status Messages** window at the bottom of the NASIS screen.

For example:

```
WHEN resultCount > 0 DISPLAY "Found %.f results" resultCount.
```

- The CVIR statements that *cannot* be used in an action script include the ACCEPT, BASE TABLE, and INTERPRET statements. Nor can anything that is related to creating a report be used in an action script.

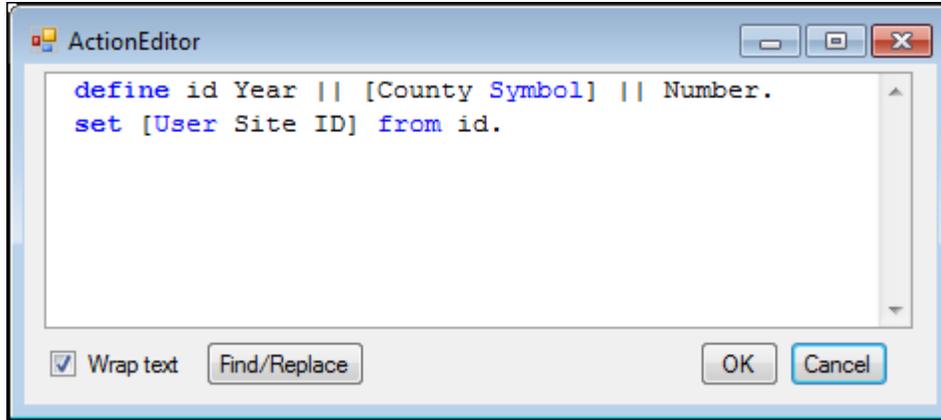
Examples of action scripts:

1. **Derive a User Site ID from its constituent parts:** A typical convention for the User Site ID field is to combine the year, state code, county code and a unique pedon number. By placing these separate fields on a form you can reduce data entry errors and make a lookup list available for the state/county portion. A form for this example looks like this:

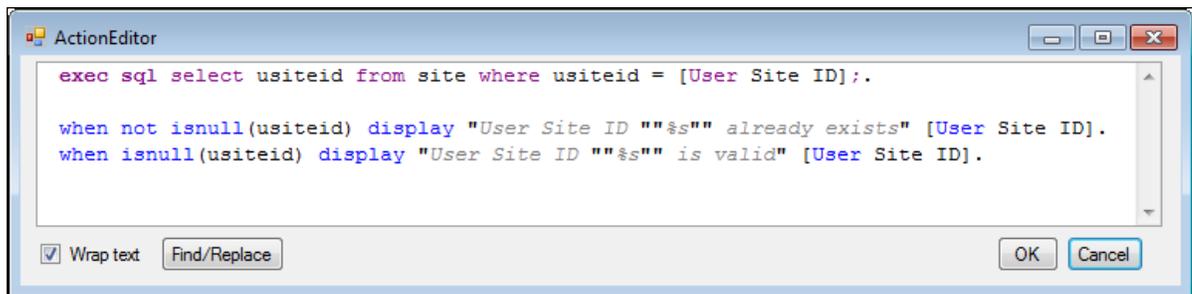


The screenshot shows a form with three tabs: 'General', 'Form', and 'Text'. The 'Form' tab is active. It contains three input fields: 'Year' with the value '2015', 'County Symbol' with a dropdown menu showing 'CO049', and 'Number' with the value '013'. To the right of these fields is a button labeled 'Set User Site ID'. Below these fields is a text field labeled 'User Site ID' containing the value '2015CO049013'.

- In NASIS the county code includes the state, so it's only necessary to use a county field. This is found in the Site table by expanding *county_iid_ref* and adding *area_symbol* to the form.
- Since the year and pedon number are not in the Site table you can create blank fields and assign their Text properties to "Year" and "Number" as shown.
- Add a button and set its Text to "Set User Site ID" (or something similar). Then use the Edit Script option to create this script:



2. **Use a query and message in an action script:** We can add a second button to the above form to check if the newly created User Site ID is unique in the local database. This requires a query in the Site table. If a matching record is found a message will be displayed in the Status window. Also, a different message can be displayed if no duplicate is found. The script would be:



```
exec sql select usiteid from site where usiteid = [User Site ID];.
when not isnull(usiteid) display "User Site ID ""%s"" already exists" [User Site ID].
when isnull(usiteid) display "User Site ID ""%s"" is valid" [User Site ID].
```

The form with a status message might look like this:

