Creating Forms

In This Chapter

- Overview
- Creating a Form
  - New Form Wizard
- Editing a Form
- Saving a Form

Overview

The basic user interface object in Servoy is a form. Forms are also a unit of scope in Servoy, meaning that forms can have variables and methods attributed to them. For more information about scope, see Implementing Business Logic.

Creating a Form

To create a form, choose one of the following ways:

- Right click on the active solution in **Solution Explorer**, select **Create New Form**
- Right click on the **Forms** node under the solution (or module) where the form needs to be created, and select **Create new form** from the context menu
- Click on the **Forms** node under the solution (or module) where the form needs to be created, and then click the **Create new form** button on the lower part of the **Solution Explorer**
- Click on the **New Form** button in the Servoy Developer toolbar

Any of these actions will open the **New Form Wizard** that will go through all the steps needed for creating a form.

New Form Wizard

To walk through the options available in the **New Form Wizard**, these instructions will use the `example_data` connection and the `customers` table.

1. Before entering a name for the form, select the table desired. Clicking on the **...** button will open a tree view to select the table from all the existing database connections. If the form won't be based on a table, select `-none-`. In the given example, select `example_data.customers`.
2. Notice that the name of the table will appear in the name field. This name can be edited; keep in mind that no two forms can have the same name in the solution. For our example, name this form 'customers_dtl'.

   > **TIP**
   >
   > Naming conventions can help one keep track of forms and tell what the form purpose is just by its name.

3. If a style sheet is being used, the name of the desired style can be selected.
4. Verify the solution name is the correct solution for the form.
5. Click **Next.**
6. Select data providers to appear on the form. These data providers can be columns, calculations, form variables, global variables, and data providers by relation. Holding the **Shift** key will select a range; holding the **Control** key will allow for multiselect. For this example, select a few columns from the table.
7. There are options on how the data providers will be added to the form. For the most part, for a regular Record View form, only the **Place with Labels** option checked is needed. For this example, just select **Place with Labels.**

   > **NOTE**
   >
   > Forms normally are created in Record view, but if one selects **Fill text property** and **Place horizontal**, the form will be set in Table view.

8. Click **Finish**.
9. The newly created form is now loaded in Form Designer.

Editing a Form

To edit a pre-existing form in Form Designer:

1. Select the form in **Solution Explorer**.
2. Do one of the following
   a. Press **Ctrl-Shift-a**
   b. Right click on the form name (or any child node of the form in the tree) and select **Open in Form Designer**.

    c. Right click on the form name in the **Solution Explorer** list view and select **Open in Form Designer**.
3. The form should appear in the Form Designer editor view on the center of the Workbench.

To edit a form in Script Editor:

1. Select the form in **Solution Explorer.**
2. Do one of the following
    a. Press **Ctrl-Shift-z**
    b. Right click on the form name (or any child node of the form in the tree) and select **Open in Script Editor**.
    c. Create a new form method. See Implementing Business Logic for more details.
3. The form should appear in the Script Editor view on the center of the Workbench.

> **NOTE**
>
> Right clicking in either editor view will display a menu that has the option to open the form in the other editor view.
> **Example**: when a form is open in Form Designer, right click and select **Open in Script Editor** from the context menu to see the form in Script Editor.

Saving a Form

Saving a form is the same as saving any other editor view in Servoy Developer, see Intro to Servoy Developer