Altia Design Tutorial

Using Altia Design to Build an Interface

1.0 Introduction

The **Altia Design** package includes an editor, runtime engine and numerous libraries of components for quickly creating user interfaces. In addition to using supplied component libraries to create graphical front ends, the Altia Design product allows users to make modified versions of existing components and create their own components in the editor without programming. With these features, a modeler can quickly create a user interface prototype for product simulations that looks and behaves like the product's real user interface.

The goal of this tutorial is to become familiar with the basic features and abilities of Altia Design. In this tutorial, we will discuss every step necessary to develop a simple interface that has animation, stimulus, control logic, connections, properties and more.

2.0 Tutorial Overview

Before you begin this half-hour tutorial, make sure you have Altia Design installed on your machine. If you need a new copy of Altia Design, please call Altia at **(719) 598-4299** or visit our web site at www.altia.com.

- The first section, labeled *Draw and Animate a Simple Meter*, will step through the procedures required to build a reusable meter complete with connections and properties.

- The next section (*Draw and Animate Two Simple Buttons*) shows how to create two buttons that will drive the meter.
- The process of connecting the buttons to the meter will be addressed in the next section, *Connect the Button to the Meter*.
- The *Tweaking the Meter* section will show you how to increase the functionality of the meter.
- Finally, the last section (*The Easy Way*) tells how to duplicate the button/meter interface using pre-built Altia components.

3.0 Draw and Animate a Simple Meter

1. Open Altia Design by choosing the **Altia Editor** icon from the **Altia Design** program group. UNIX users open Altia Design by executing `altia` from the Altia Design software `bin` directory.
2. Our simple meter will consist of a vertical indicator or needle that moves horizontally over a rectangle. Let's start by drawing the rectangular "backplane" of the meter. Click on the Rectangle drawing tool in the Tool-Command panel.

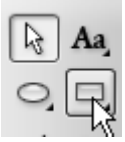

FIGURE 3-1 Rectangle Tool Button

3. In the drawing area, press the left mouse button and drag a large, horizontal rectangle then release the mouse button. To change its color, left click on the FG color rectangle in the Tool-Command panel to open the color picker, choose a light grey color and then click the **OK** button. When you are done, your rectangle should resemble the one shown in Figure 3-2. Notice the "handles" around the rectangle which indicate that it is the selected object.

FIGURE 3-2 Our First Rectangle

4. Now, let's draw the "needle" of our meter. With the Rectangle tool still selected, draw a thin, vertical rectangle on top of the backplane rectangle (make the nee-

dle slightly taller than the backplane and position it on the left end). Change the needle color to red by clicking on the FG rectangle and choosing that color in the color picker. When you are done, your meter should look something like the one in Figure 3-3.

FIGURE 3-3 The Basic Meter Layout

5. Before animating the needle, let's group it so that our meter can be easily modified later on (we'll discuss this further in Section 6.0, *Tweaking the Meter*). Choose the Selection Tool in the Tool-Command panel. Then, to put the needle into a group, just select it (if it is not already selected) and then press the **Group** button near the bottom of the Tool-Command panel.

6. Now, let's animate the group that contains our meter's needle. Open the Animation Editor by clicking on the **Animate** button located above the drawing area.

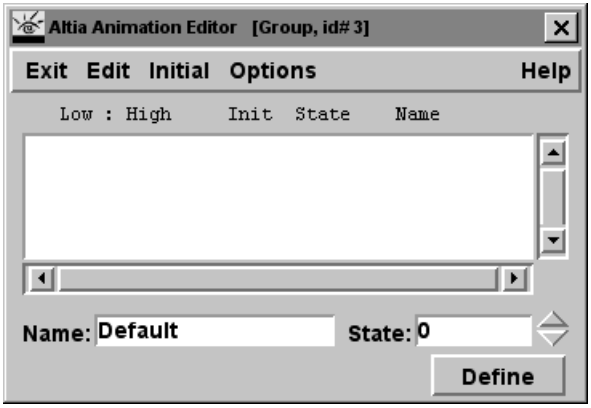

FIGURE 3-4 The Animation Editor

7. With the needle group still selected, type `meter` in the Animation Editor's **Name** field and press Enter. Click the **Define** button to define state 0 of our `meter` animation.

8. Next, move the needle to the right side of the backplane by holding down the Control key and then pressing the left mouse button on the needle and dragging it to the right. Release the mouse button when the needle is properly positioned. This will be our highest meter position.

9. In the Animation Editor, change the **State** field to 10, press Enter and then click **Define**. We have now defined an animation named `meter` from states 0 to 10 for the needle group as shown in Figure 3-5. Try changing the `meter` animation to a value between 0 and 10 by using the decrement button to the right of the State field. Pretty neat, huh?

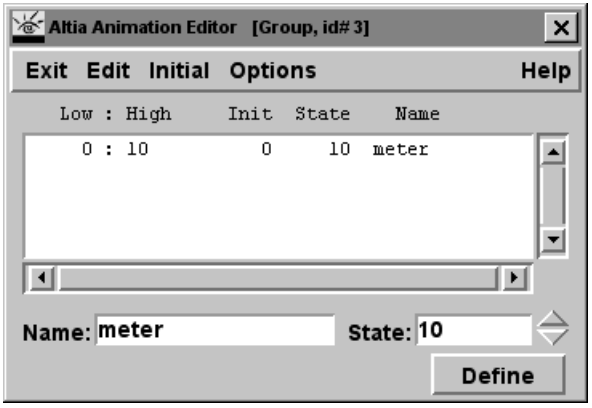

FIGURE 3-5 The meter Animation

10. Even though our meter now animates, it isn't very useful without labels. We will add two different kinds of labels to this meter: one static and the other dynamic.

11. For variety's sake, we will make the left-most label static. To add this label, click on the Text Tool button in the Tool-Command panel then click in the drawing area below the left side of the meter.

FIGURE 3-6 Text Tool Button

12. Type `0` and then press Enter. Change the text color to black by choosing that color using the color picker.

13. For the right-side label we will use a dynamic text object from the **`read-outs.dsn`** component library.

14. To open a component library, click the **Libraries** button above the drawing area.

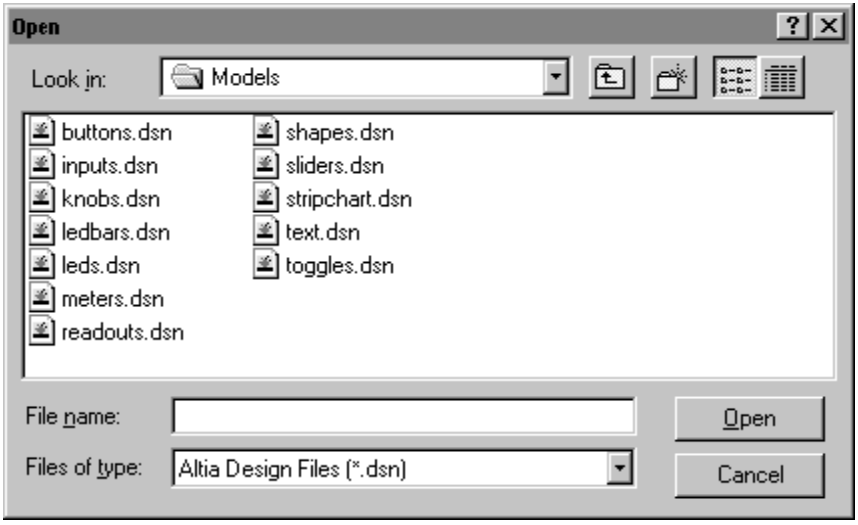

FIGURE 3-7 Open Models Dialog

15. Choose the **`readouts.dsn`** library and then click **Open** to view a collection of pre-built readouts. UNIX users must first double-click on the `$ALTIAHOME/models/` entry in the **Standard Models Directories** list to go to the directory that contains the **`readouts.dsn`** file.

16. Pull a readout (whichever one appeals to you most) into your design by pressing the left mouse button on it and dragging it into the Altia Design window. Close the Libraries view.

17. You may notice that the readout font differs from the static text font. This is easily remedied. Select the text tool and then, with the readout still selected, choose a new font from the drop down.

FIGURE 3-8 Font Selection Drop Down

18. Since this readout will only be displaying integer values, let's modify its properties to suit our design. Select the Altia Design main window, make sure the Selection Tool is selected and double-click the readout to open its **Property Dialog**.

19. Change the **Current Readout** property from **0.00** to **10** and the **Decimal Digits** property to **0**. Close the Property Dialog.

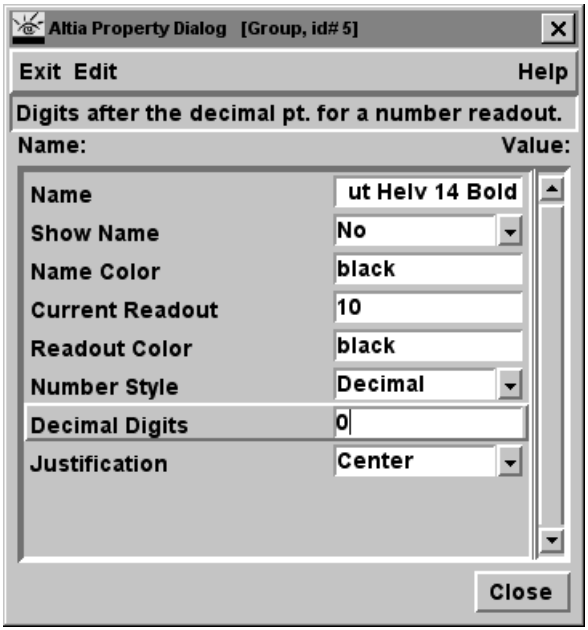

FIGURE 3-9 Readout Property Dialog

20. Note the animation name of the readout in the Animation Editor (e.g., `5_readout_float`). We will use this readout animation name later in this section.

21. Our meter has all of its components now. Let's group these components together to make a single meter object. To do this, click and drag around all of the items (the backplane, the needle and the two labels) and then press the **Group** button.

22. In order for other Altia objects to easily connect to our meter, let's add a connector to it. With the meter group still selected, choose **Selected Objects** from the **Connections** menu.

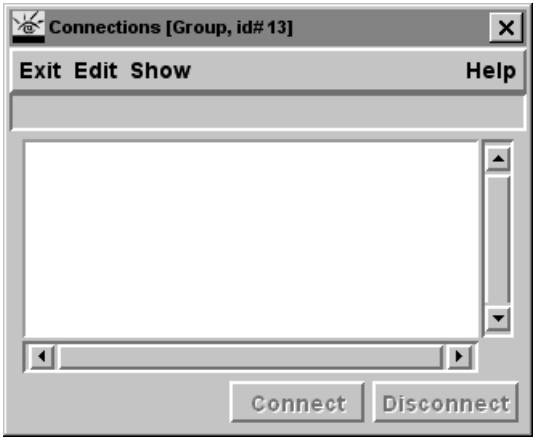

FIGURE 3-10 Empty Connection Dialog

23. From the Connections dialog's **Edit** menu, choose **Add Connection...** to bring up the EditConnection dialog. In the **I/O Name** field, type `My Meter Value` and in the **Animation Name** field, type `meter`. Click OK.

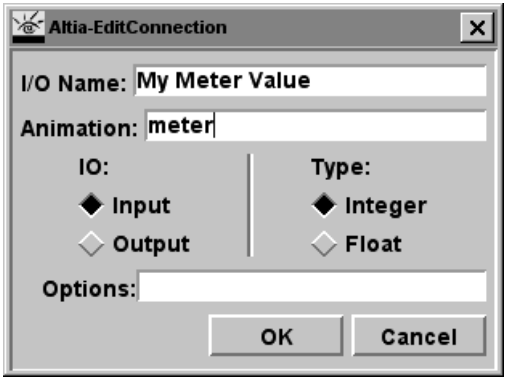

FIGURE 3-11 Edit Connection Dialog

24. We have now created a connector for our meter's `meter` animation. Any Altia object that has a connector (such as those in the component libraries) can now easily drive our meter.

25. The next step is to create a property for our meter that allows us to change the maximum display value. Note that this property will actually be changing the readout label *and* the animation range of the meter.

26. Open the Property Dialog by choosing **Set Properties...** from the **Edit** menu of the Altia Editor. From the **Edit** menu of the Property Dialog, choose **Add Property...**.

27. In the Property Name dialog that appears, type `Maximum` then click **OK**.

28. In the Current Specification section of the Altia Property Editor dialog, click the **Animation** radio button and choose the **Range** option on the button to its right. In the **Animation Name** field at the bottom, type in `meter` and then click the ra-

dio button beside the word **Max** as shown below in Figure 3-12. Click the **Define** button to add this specification to the property.

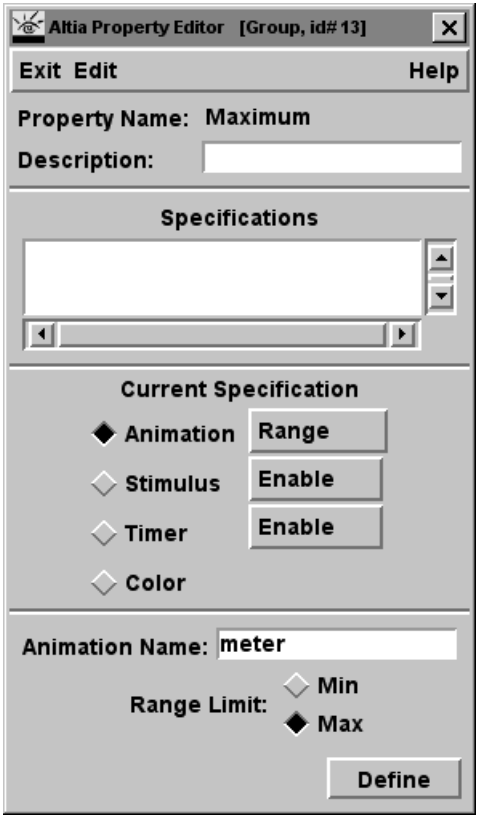

FIGURE 3-12 Set Property to Change Meter's Animation Range

29. This property will change the maximum animation value of our meter allowing it to display a wide range of values. As an added feature, let's set up this property so that it also changes the right side label to accurately reflect the capability of our meter.

30. In the Current Specification section of the Altia Property Editor dialog, change the button to the right of **Animation** from **Range** to **Value**. In the **Animation Name** field at the bottom, type in the animation name for the readout object (the name that you have been remembering since step 20) as shown below in Figure 3-13.

Click **Define** to add this specification to the property and then close the Altia Property Editor dialog.

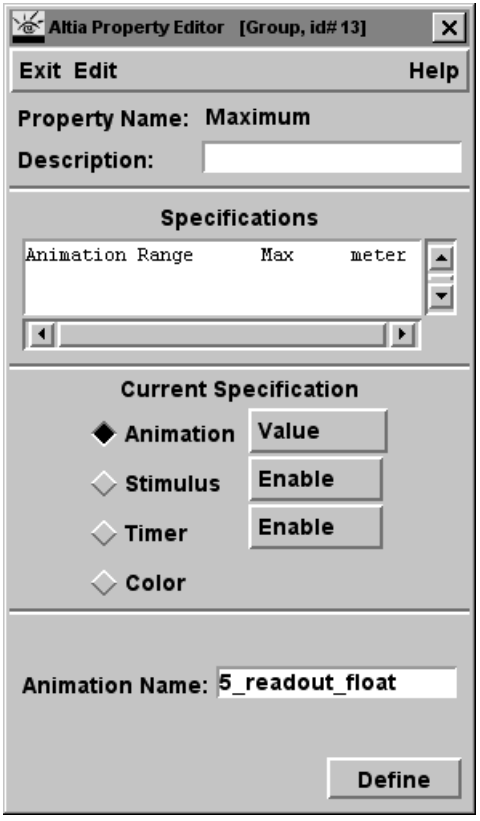

FIGURE 3-13 Set Property to Change Label

31. Now we have a complete, reusable meter of which we can be proud. Next, let's create some buttons to drive it.

4.0 Draw and Animate Two Simple Buttons

1. Use the Rectangle tool from the Tool-Command panel to draw a small black square in the drawing area below the meter. In the Animation Editor, type `button` in the **Name** field, `0` in the **State** field, press Enter and then click **Define**.

2. Go back to the Altia Editor's drawing area and, with the button you've drawn still selected, use the cursor keys to "nudge" the box 3 pixels to the right and 3 pixels down (push the right arrow key 3 times, then push the down arrow key 3

times). Change the color of the box to red using the FG color rectangle to open the color picker.

3. This will be state 1 for our button. To define this new animation state, just change the **State** value in the Animation Editor to 1, press Enter and then click **Define**. Test the new 0 and 1 states of the `button` animation by using the up/down arrow buttons next to the **State** field.

4. Next, we have to define the input stimulus to which our button will listen. Press the **Stimulate** button located above the drawing area. We will use this editor to define what happens to our `button` animation on a left mouse button down or left mouse button up.

5. In the **WHEN INPUT STIMULUS IS** section (the top-most section) of the Stimulus Editor, click on the left mouse button graphic to change the input stimulus to LeftDown (see Figure 4-14).

6. In the next section (**IN THE AREA**), click the **Whole** radio button to indicate that we want our button to be sensitive to left clicks over its entire area (see Figure 4-14).

7. In the **AND THE ENABLE CONDITION** section, type `button` in the first field, click on the relationship button (initially labeled with "**=**") to choose **<>** and then put `1` in the value field (see Figure 4-14).

8. Lastly (for this stimulus definition, anyway), in the **THEN EXECUTE** section, type `button` in the first field and then put `1` in the state field. Click the **Define** button (see Figure 4-14).

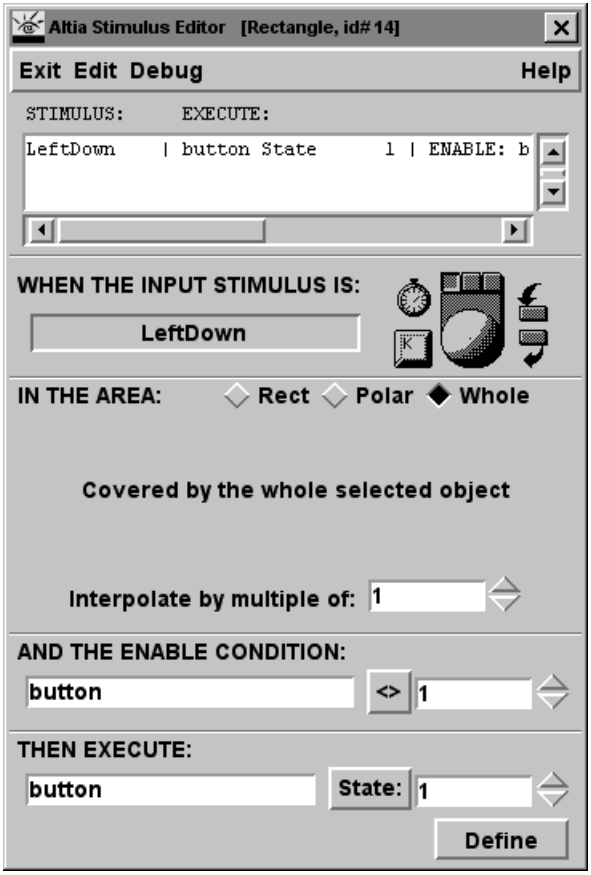

FIGURE 4-14 LeftDown Stimulus Definition

9. Next, let's create a LeftUp stimulus to tell our button when to release. In the Stimulus Editor, change the stimulus type to **LeftUp** by clicking on the left

mouse button graphic again. Change the enable condition to be **button <> 0** and the execute statement to be **button State: 0**. Click the **Define** button.

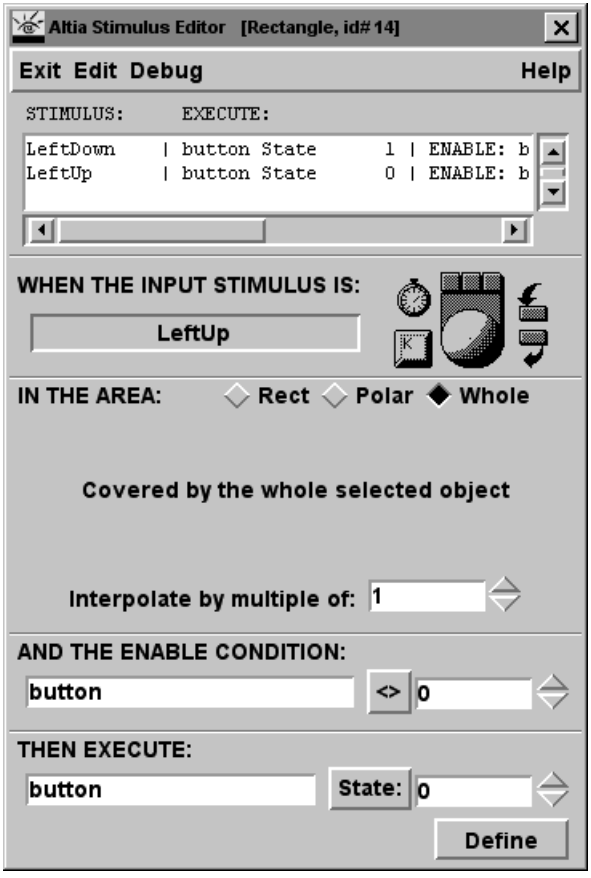

FIGURE 4-15 LeftUp Stimulus Definition

10. Close the Stimulus Editor. Click the Run toggle button located directly beneath the **Object** menu. Now, click and release our button a few times.

11. In order to cover all the bases, let's use some control code to implement a counter that increments whenever the button is pressed. We want to add the control code to the button, so the button must be the selected object. If it is not the selected object, change back to Edit mode and right-click the button object to select it. Click the **Control** button located above the drawing area to open the Control Editor.

12. In the Control Editor, click the **Add...** button and then choose to add the **WHEN** statement. In the first field of the **WHEN** block, type `button`. In the last field,

type `1`. We are telling this control logic to execute whenever the value of `button` is set to 1 (whenever we press our button).

FIGURE 4-16 WHEN Statement

13. Click the **Add...** button again and choose the **EXPR** (expression) statement. Put `button_count` in the first field and `button_count+1` in the last field. This statement will increment a variable called `button_count` by 1. The `button_count` variable is created automatically simply by using it in control. Its initial value is 0.

FIGURE 4-17 EXPR Statement

14. That's all for the control logic. Now let's add an identifying label beside this button using the static text tool. First click the Edit toggle button to return to Edit mode if you are in Run mode. Then, just click on the Text Tool button, click to the right of the button, type `Count` and press Enter.

15. Let's make a copy of this button (and all of its functionality) that we can use to reset our counter.

16. Choose the Selection Tool, select our Count button and its label, press **Ctrl+C** then **Ctrl+V** to copy and paste them. Next, move the copies below the original button.

17. To edit the static text label, simply double-click on the `Count` text, edit the string to read `Reset` (use the Backspace key to delete existing characters) and then press Enter.

18. Select the Reset button next. You can click on it with the right mouse button to select it exclusively when multiple objects are already selected. From the Animation Editor's **Edit** menu, choose the **Rename Animation...** option. In the Function Rename Dialog that opens, choose the `16_button_count` animation (the numeric prefix may be different for you) in the list at the top. In the third line below

the list box (which begins with **Rename function**), type `button_count` in the *second* field and then press Enter.

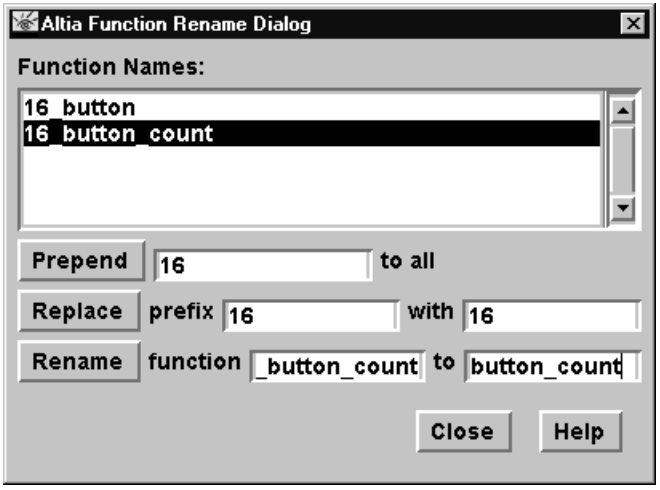

FIGURE 4-18 Rename Function Dialog

You should notice that the animation `16_button_count` in the list box changes to `button_count`. Close the Function Rename Dialog.

19. To complete this button, we need to edit its control to reset `button_count`.

20. Go back to the Control Editor (which should still be open). Click the **WHEN** block to expand it and then change the **EXPR** such that it sets `button_count` to 0.

FIGURE 4-19 Reset EXPR Statement

21. Group the two buttons and their labels together by selecting them and pressing the **Group** button.

22. Now, let's add a connector to our button group's `button_count` variable. Note that we will be making a connector for a variable that is used in control whereas for the meter we made a connector to one of the object's animations.

23. In the Altia Editor, select our button group and then choose **Selected Objects...** from the **Connections** menu. From the Connections dialog's **Edit** menu, choose **Add Connection...** to bring up the EditConnection dialog. In the **I/O Name** field, type `Button Counter` and in the **Animation** field, type `button_count`. Choose the **Output** radio button and then click **OK**.

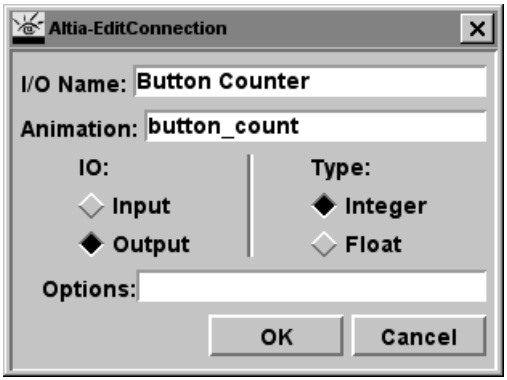

FIGURE 4-20 Button Counter Connector

5.0 Connect the Button to the Meter

1. At this point, the Connections dialogs for the buttons and meter should both be open. If they are not, then just select the meter and the buttons and then choose **Selected Objects** from the **Connections** menu.

2. To connect the buttons to the meter, select the `OUTPUT Button Counter` connector in the button group's Connections dialog. Next, select `INPUT My Meter Value` from the meter's Connections dialog.

3. When the two signals are selected, the **Connect** button in each dialog becomes available. Click either one to connect the buttons to the meter then close the Connections dialogs.

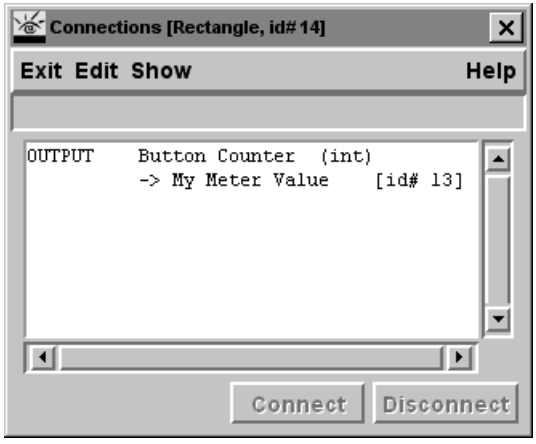

FIGURE 5-21 Connected Components

4. Now, let's give our design a little test run. *Select the meter* and then put the editor into Run mode by choosing the Run toggle located directly beneath the **Object** menu.

5. Click the Count button several times to increment the value feeding the meter. Press the Reset button to set the meter to 0.

6. Now try changing the **Maximum** property of the meter using the Property Dialog. Notice that when you change the **Maximum** property, you change the number of animations for the meter *and* the right side label. This is also shown in the Animation Editor's list area because the meter is the selected object.

7. When your meter and button groups are finished, they should look something like Figure 5-22.

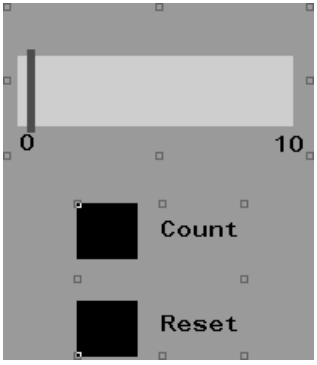

FIGURE 5-22 Finished Design Layout

6.0 Tweaking the Meter

1. To make our meter a little more informative, let's add a display that shows the current meter value to supplement the needle's position. Return to Edit mode by choosing the Edit toggle located directly beneath the **File** menu.

2. Recall that our meter is a group containing several objects, one of which is the group containing the needle. Now we'll explore the power of grouped objects. Select the meter group and then click the **FocusIn** button near the bottom of the Tool-Command panel. Don't be alarmed that the objects outside of this focus level (the buttons) disappear. That just helps us concentrate on the items in the group.

3. Now, right click on the group containing the needle to select it. Note that we use a right mouse button click to select the needle when multiple objects are already selected. Click the **FocusIn** button again.

4. Open the readout component library again. Drag in a readout and place it directly above the needle. Open its properties and change the **Decimal Digits** to 0. Switch to the Text Tool and change the font to match that of the other meter labels.

5. In order to make sure that this label always shows the current value of the meter, we will add control logic that sets its "integer" animation whenever the meter value changes. Note the name of the integer animation in the Animation

Editor (e.g., `16_readout_integer`) and then open the Control Editor by clicking the **Control** button located directly above the drawing area.

6. In the Control Editor, click the **Add...** button and then choose to add the **WHEN** statement. In the first field of the **WHEN** block, type `meter`. Leave the last field empty. We are telling this control logic to execute whenever the value of `meter` changes.

FIGURE 6-23 WHEN Statement

7. Click the **Add...** button again and choose the **EXPR** (expression) statement. Put `16_readout_integer` (or whatever your readout's integer animation was) in the first field and `meter` in the last field.

FIGURE 6-24 EXPR Statement

8. Click the **FocusOut** button two times so that we are back to the focus level where all objects are visible.

9. Go back into Run mode and press the buttons a few times. Your finished button / meter interface should resemble the following.

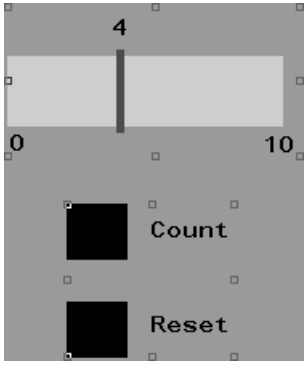

FIGURE 6-25 Tweaked Design Layout

7.0 The Easy Way

1. Although it is a good idea to understand how to build Altia components, much effort can be saved by using the libraries of pre-built Altia components. As a quick illustration, let's duplicate our button/meter interface.

2. Click the **Libraries** button and open the **buttons.dsn** component library.

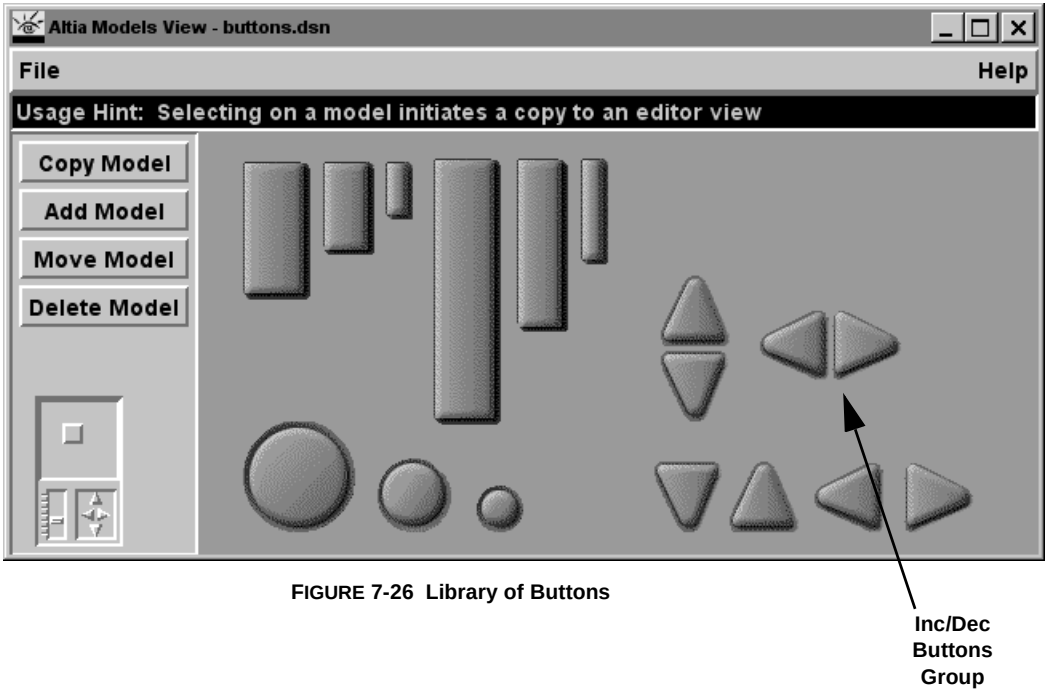

FIGURE 7-26 Library of Buttons

3. Scroll up to find the Inc/Dec buttons indicated in Figure 7-26 and drag them into the Altia drawing area.

4. Open the **meters.dsn** library and drag in a needle gauge.

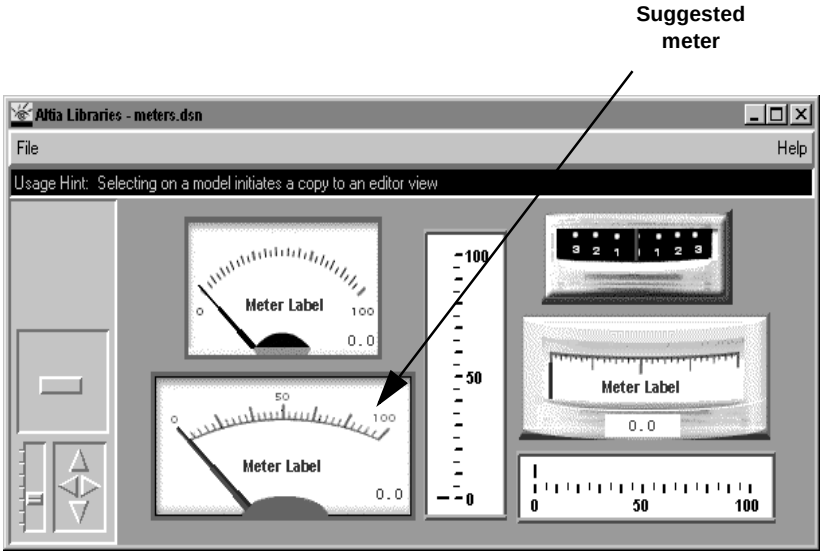

FIGURE 7-27 Library of Meters

5. Open the Connections dialogs for the new library versions of the buttons and meter. Connect the `OUTPUT Inc/Dec Value` signal to the `INPUT Meter Value`.

6. Switch to Run mode and click the Inc/Dec buttons.

7. Now, let's save our GUI to a design file. In the Altia Design window, choose **Save** from the **File** menu. Save your design to the file **easy.dsn**.

8.0 Tutorial Summary

In this tutorial, we began by creating an Altia meter with connections and properties. Next, we built two simple buttons and connected them to our meter using Altia Design's connection dialogs. We then further tweaked our meter through the power of grouped objects. Finally, we explored how to save time and effort using Altia's prebuilt component libraries.