Polymorphic Units in LabVIEW™

Introduction

Using units gives you an additional level of consistency checking when evaluating expressions and formulas, also known as dimensional analysis. With LabVIEW, you do not have to worry about conversions between systems of units, because LabVIEW handles unit conversion when it displays or enters data.

The LabVIEW built-in functions, such as Add and Multiply, are polymorphic with respect to units, so they automatically handle different units. But in order to build a subVI with the same polymorphic unit capability, you must use polymorphic units.

You can use polymorphic units for one VI to do the same calculation, regardless of the units received by the inputs. For example, if you want to create a VI that computes the root mean square value of a waveform, you must define the unit associated with the waveform. A separate VI is necessary for voltage waveforms, current waveforms, temperature waveforms, and so on. However, instead of rewriting the same VI for each case, you can write a single subVI with polymorphic units and call it from the VIs with specific units.

Creating Polymorphic Units

Create a polymorphic unit by entering `$`*x* in the unit label of a numeric control or indicator on the front panel, where *x* is a number, for example, `$1`. You can think of `$`*x* as a placeholder for the actual unit. When the VI is called, LabVIEW substitutes the units you pass in for all occurrences of `$`*x* in that VI.

If you need to use more than one polymorphic unit, you can use the abbreviations `$2`, `$3`, and so on.

Complete the following steps to create a polymorphic unit in your VI.

1. Right-click a numeric object on the front panel and select **Visible Items»Unit Label** from the shortcut menu.
2. Type `$`*x*, where *x* is any number, 1 through 9.
3. Click outside the numeric object on the front panel to end the editing session.

Using Polymorphic Units

LabVIEW treats a polymorphic unit as a unique unit. It is not convertible to any other unit and propagates throughout the diagram just as other units do. When you wire a control with the polymorphic unit `$1` to an indicator that also has the polymorphic unit `$1`, the units match and the VI can compile.

`$1` can be used in combinations like any other unit. For example, if the control is multiplied by 3 seconds and then wired to an indicator, the indicator must be `$1 s` units. If the indicator has different units from the control, the block diagram shows a broken wire.

A call to a subVI containing polymorphic units computes output units based on the units received by its inputs. For example, suppose you create a subVI that has two inputs with the polymorphic units `$1` and `$2` that creates an output in the form `$1$2/s`. If the subVI is wired with inputs of `m/s` to the `$1` input and `kg` to the `$2` input, the output unit is computed as `kg m/s^2`.

Suppose a different VI has two inputs of `$1` and `$1/s` and computes an output of `$1^2`. If this VI is wired with inputs of `m/s` to the `$1` input and `m/s^2` to the `$1/s` input, the output unit would be computed as `m^2/s^2`. However, if this VI is wired with inputs of `m` to the `$1` input and `kg` to the `$1/s` input, the subVI call is broken. One of the inputs is declared to be a unit conflict, and the output would be computed (if possible) from the other. A VI with polymorphic units can have a subVI with polymorphic units because the respective units are kept distinct.

When to Use Polymorphic Units

This section presents several examples of VIs with and without polymorphic units. We begin with a simple averaging example without any units. Then we provide an example with a base unit. We turn this example into one with polymorphic units and use it as a subVI for another example. The *Multiple Averages* example and the *Multiple Rates* example demonstrate when you might want to use a subVI with polymorphic unit capability.

Averaging without Units

Figure 1 shows the front panel of a VI that computes the average of two numbers. Notice that there are no units on the controls or indicator.

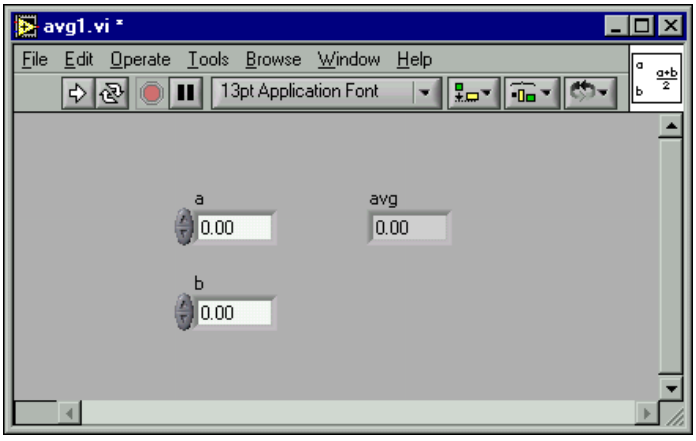

Figure 1. Simple Averaging Front Panel

Now take a look at the block diagram, shown in Figure 2.

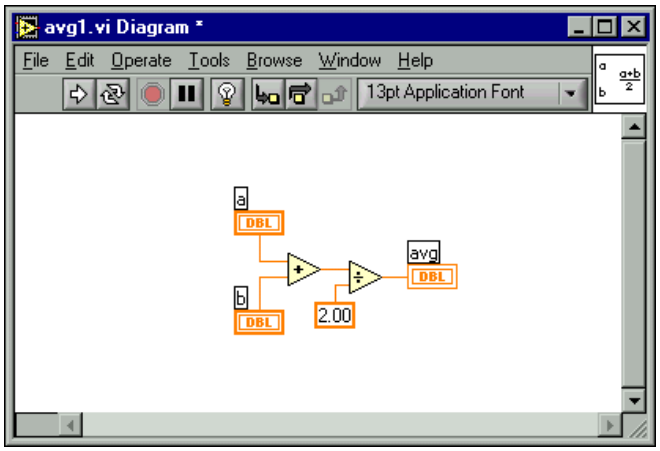

Figure 2. Simple Averaging Block Diagram

Averaging with One Unit

Figure 3 shows the front panel of a VI that computes the average of two numbers, each with a unit of meters.

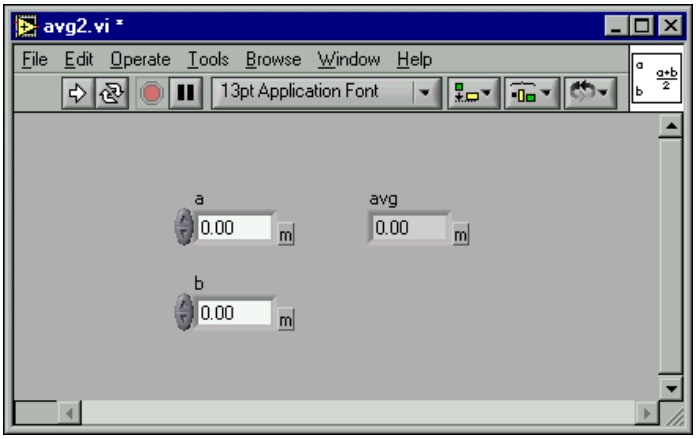

Figure 3. Averaging Meters Front Panel

The block diagram for this VI is identical to the block diagram shown in Figure 2. The units appear only on the front panel. If one of the front panel controls or the indicator does not have a unit and the other front panel objects do, the wires appear broken on the block diagram, and the VI does not run.

Averaging with Polymorphic Units

Figure 4 shows the front panel of a VI that computes the average of two numbers. Unlike Figure 3, Figure 4 has a polymorphic unit. We use this VI as a subVI in the *Multiple Averages* example later in this application note.

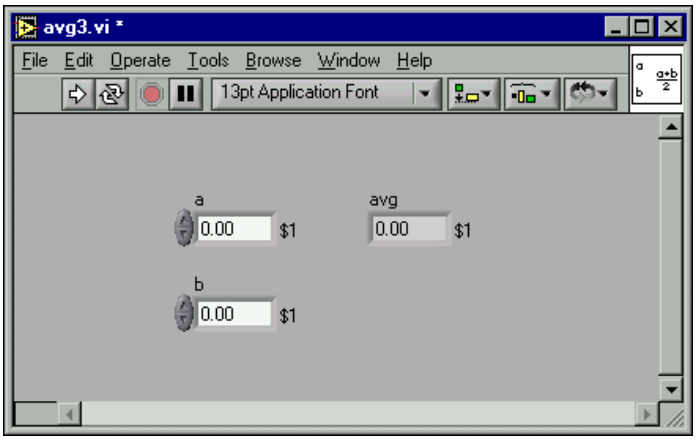

Figure 4. Averaging Polymorphic Units Front Panel

The block diagram for this VI is identical to the block diagram shown in Figure 2. As with the *Averaging with One Unit* example, the polymorphic units appear only on the front panel.

Multiple Averages

Now that you have seen examples of the same VI without units, with units, and with polymorphic units, we can look at an example of when to use a subVI with polymorphic units. Figure 5 shows the front panel of a VI that calculates three different averages, each with different units.

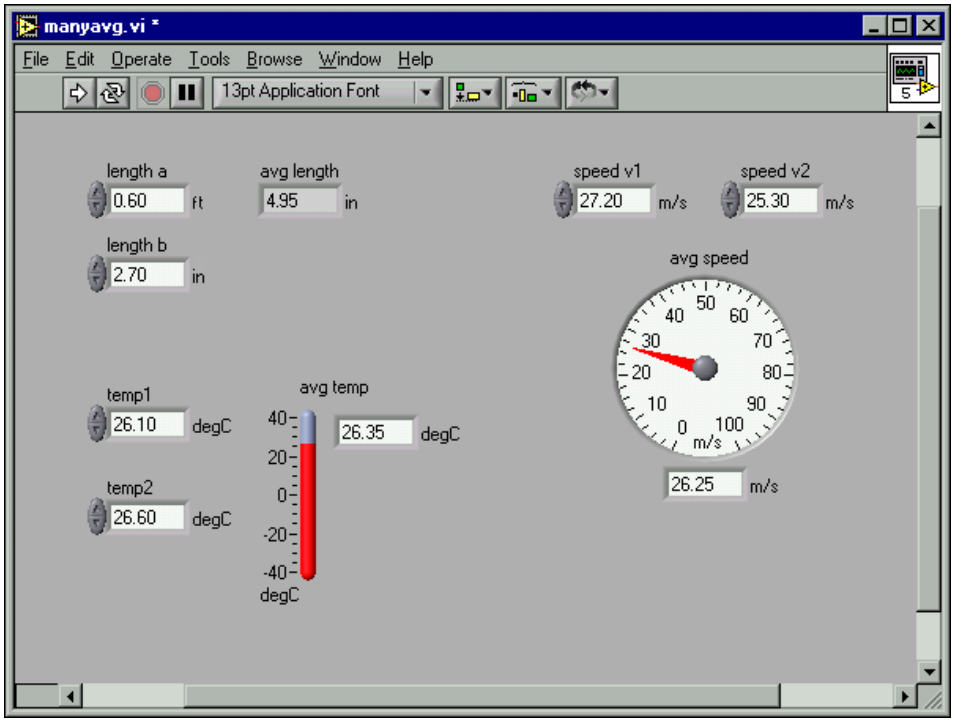

Figure 5. Multiple Averages Front Panel

Notice that **length a** has a unit of feet, **length b** has a unit of inches, and **avg length** has a unit of inches. LabVIEW automatically handles the conversion from feet to inches for you.

Now take a look at the block diagram, shown in Figure 6. We use the Averaging with Polymorphic Units VI to average each example shown in Figure 5.

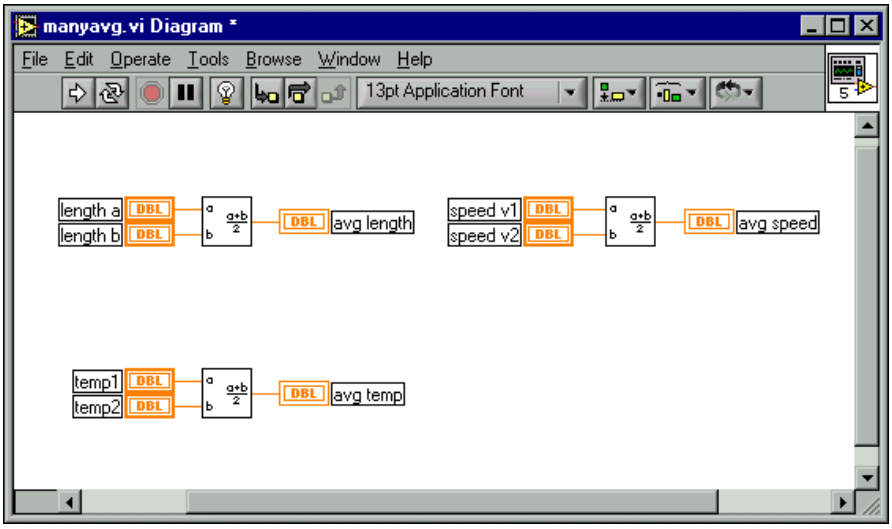

Figure 6. Multiple Averages Block Diagram

Notice that you can use the Averaging with Polymorphic Units subVI with all the units shown in Figure 5. Instead of creating a separate VI each time you want to average two numbers with units, you can use a subVI with polymorphic units.

Rates with Polymorphic Units

You also can use polymorphic units when your indicator unit is a combination of the input units, such as calculating rates. Figure 7 shows the front panel of a VI that calculates a quantity per interval. We use this VI as a subVI in the *Multiple Rates* example later in this application note.

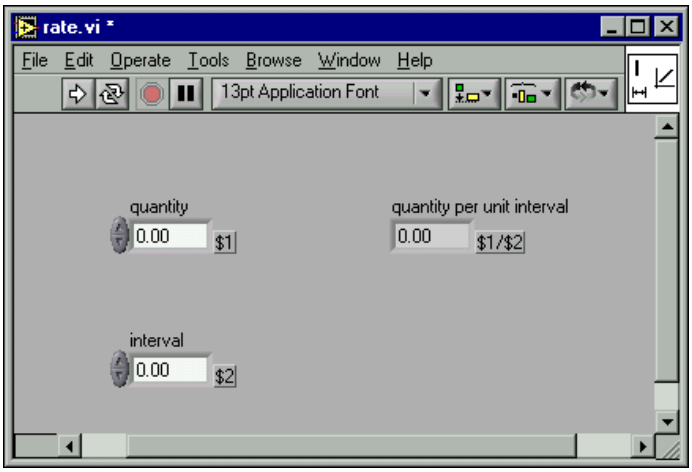

Figure 7. Rates with Polymorphic Units Front Panel

Notice that the quantity unit is `$1` and the interval unit is `$2`. You can use any number, 1 through 9, for your polymorphic unit.

The block diagram is shown in Figure 8. You usually do not create such simple subVIs. This is just an example to illustrate two polymorphic units on a subVI.

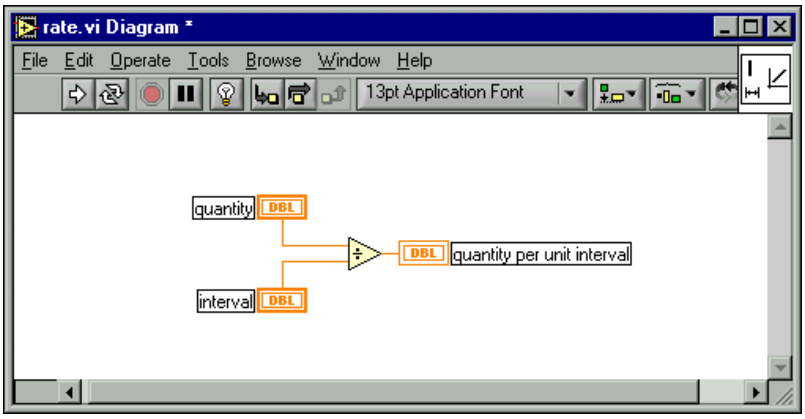

Figure 8. Rates with Polymorphic Units Block Diagram

Multiple Rates

Figure 9 shows the front panel of a VI that calculates three different rates – gradient, acceleration, and pressure.

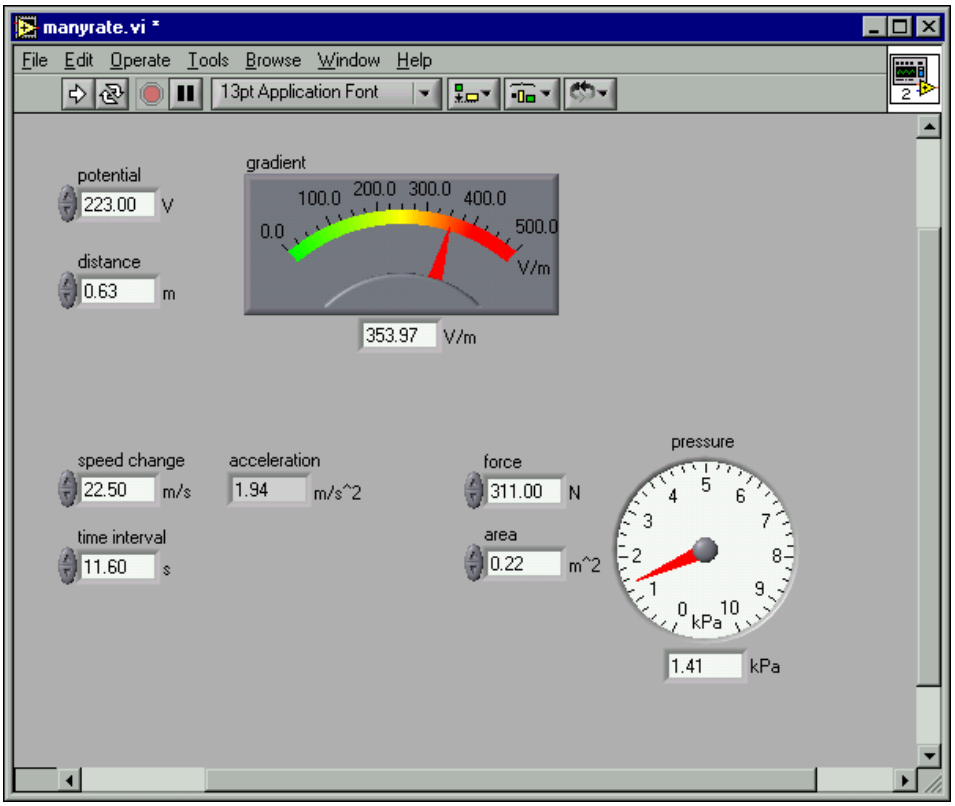

Figure 9. Multiple Rates Front Panel

Notice that two of the units on the controls are compound units. The Rates with Polymorphic Units subVI accepts both simple and compound units.

Now take a look at the block diagram, shown in Figure 10.

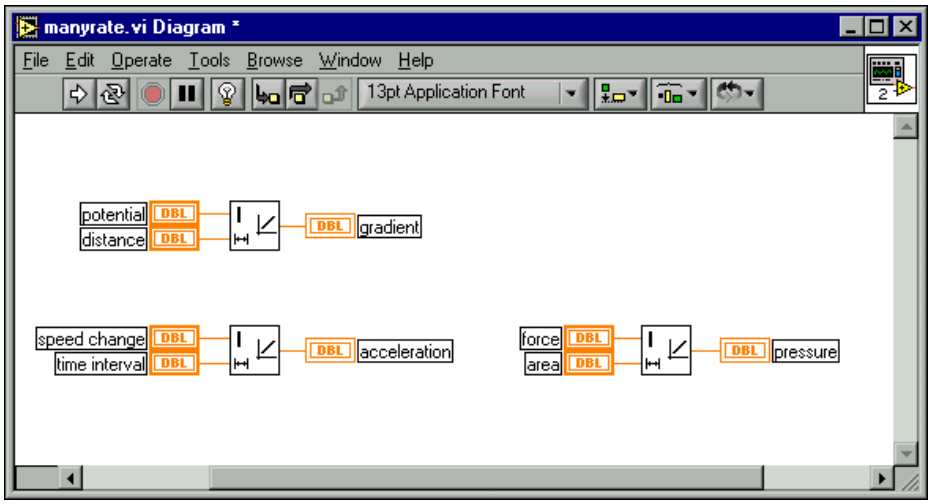

Figure 10. Multiple Rates Block Diagram

Notice that the block diagrams in Figure 6 and Figure 10 are almost the same. Each performs three different calculations using one subVI with polymorphic units.

342081A-01 Jul00