Application Note 170

Polymorphic Units in LabVIEW™

Introduction

Use units to gain an additional level of consistency checking when you evaluate expressions and formulas, also known as dimensional analysis. With LabVIEW, you do not have to worry about conversions among systems of units because LabVIEW handles unit conversion when it displays data or when you enter data.

The LabVIEW built-in functions, such as Add and Multiply, are polymorphic with respect to units and automatically handle different units. However, 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 the inputs receive. 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 a 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.

Refer to the *LabVIEW Help* for the most recent version of these instructions.

342081B-01 March 2004

Using Polymorphic Units

LabVIEW treats a polymorphic unit as a unique unit. The polymorphic unit cannot convert to any other unit and propagates throughout the block 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.

You can use `$1` in combinations like any other unit. For example, if you multiply a control by 3 seconds and wire it to an indicator, the indicator must be `$1 s` units. Wiring an indicator that has different units from the control results in a broken wire.

A call to a subVI that contains polymorphic units computes output units based on the units its inputs receive. 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 you wire the subVI with inputs of `m/s` to the `$1` input and `kg` to the `$2` input, LabVIEW computes the output unit 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 you wire this VI with inputs of `m/s` to the `$1` input and `m/s^2` to the `$1/s` input, LabVIEW computes the output unit as `m^2/s^2`. However, if you wire this VI with inputs of `m` to the `$1` input and `kg` to the `$1/s` input, the subVI call is broken. LabVIEW declares one of the inputs as a unit conflict and computes (if possible) the output from the other unit. A VI with polymorphic units can have a subVI with polymorphic units because LabVIEW keeps the respective units distinct.

When to Use Polymorphic Units

This section presents several examples of VIs with and without polymorphic units. It begins with a simple averaging example without any units. It then provides an example with a base unit. Next it turns this example into one with polymorphic units and uses 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.

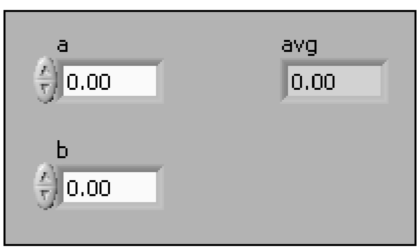

Figure 1. Simple Averaging Front Panel

Figure 2 shows the block diagram of the VI.

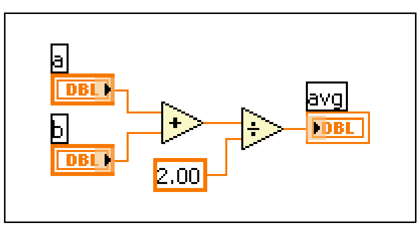

Figure 2. Simple Averaging Block Diagram

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.

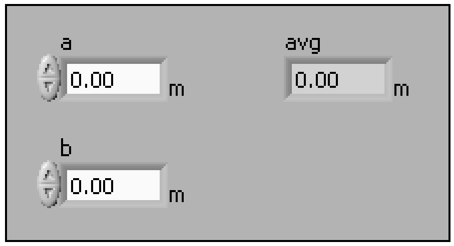

Figure 3. Averaging Meters Front Panel

The block diagram for this VI is identical to the block diagram in Figure 2. The units appear only on the front panel. If one control or indicator on the front panel has a unit, all controls and indicators on the front panel must have a unit or the block diagram displays broken wires, and the VI will 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. The Multiple Averages example later in this document uses this VI as a subVI.

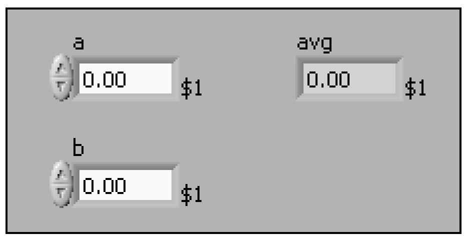

Figure 4. Averaging Polymorphic Units Front Panel

The block diagram for this VI is identical to the block diagram in Figure 2. As with the Averaging with Meters example, the polymorphic units appear only on the front panel. If one control or indicator on the front panel has a unit, all controls and indicators on the front panel must have a unit or the block diagram displays broken wires, and the VI will not run.

Multiple Averages

Figure 5 shows the front panel of a VI that calculates three different averages, each with different units, which is an example of when to use a subVI with polymorphic units.

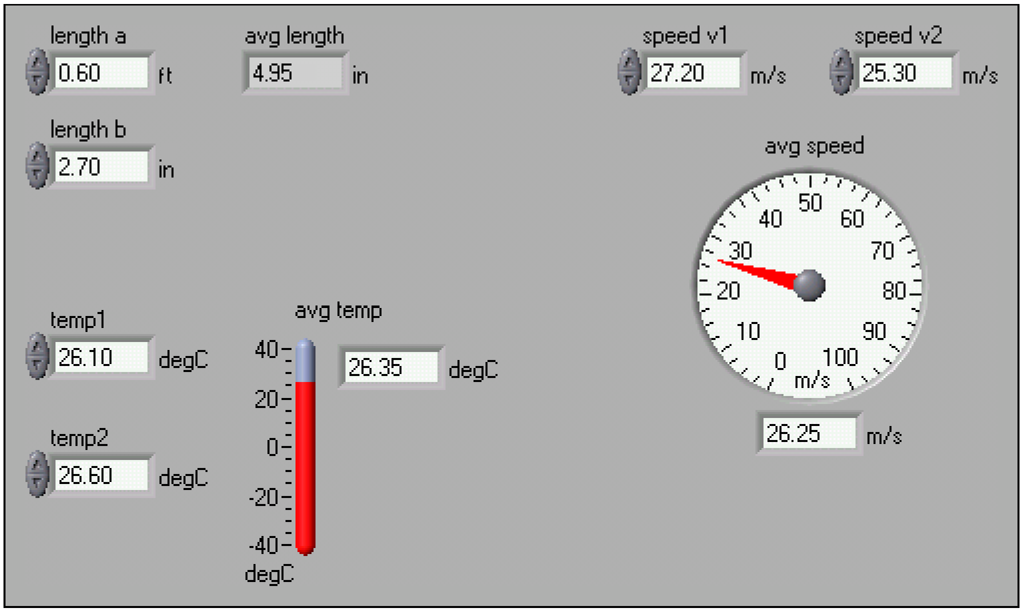

Figure 5. Multiple Averages Front Panel

Notice in Figure 5 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 because this VI uses polymorphic units.

Figure 6 shows the block diagram, which uses the Averaging with Polymorphic Units VI in Figure 4 to average each example shown in Figure 5.

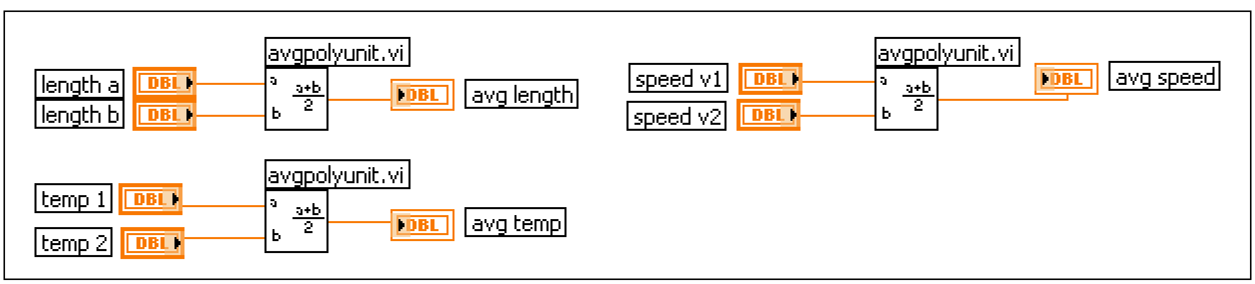

Figure 6. Multiple Averages Block Diagram

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 an 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. You usually do not create such simple subVIs. This is just an example to illustrate two polymorphic units on a subVI. The Multiple Rates example in Figure 10 uses this VI as a subVI.

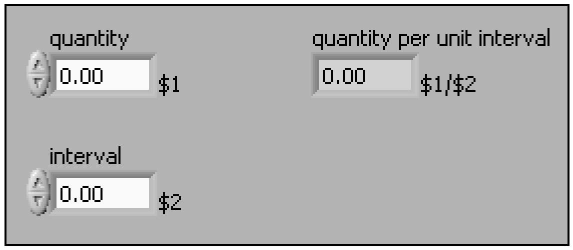

Figure 7. Rates with Polymorphic Units Front Panel

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

Figure 8 shows the block diagram of the VI.

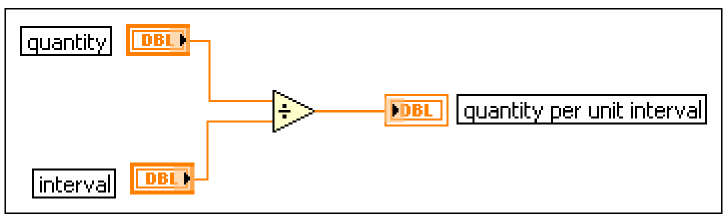

Figure 8. Rates with Polymorphic Units Block Diagram

Multiple Rates

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

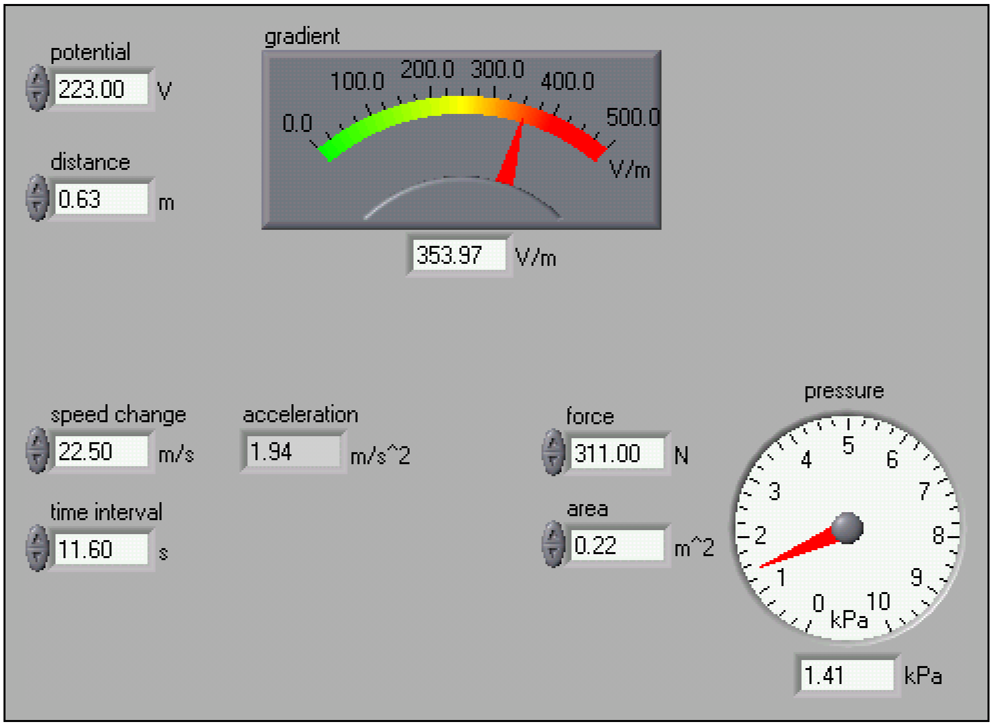

Figure 9. Multiple Rates Front Panel

Notice that two of the units on the controls are compound units. The Rates with Polymorphic Units subVI in Figure 8 accepts both simple and compound units.

Figure 10 shows the block diagram of the VI.

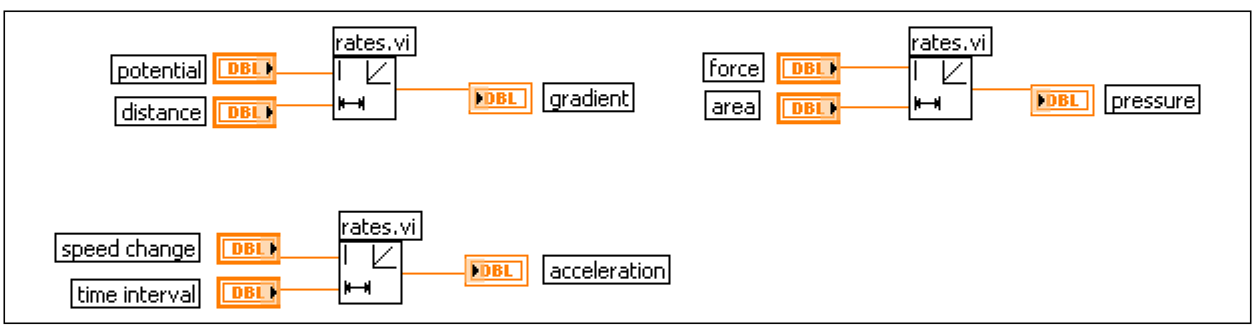

Figure 10. Multiple Rates Block Diagram

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.