

How to Use Basic Numeric Data Types Tutorial

Numeric Representations of Double, I32, and more...

There are many different data types but the most frequently used data type in LabVIEW is the Numeric data type. The most commonly used Numeric data type representations are Double (64-bit precision floating point), and Integer (32-bit integer or I32).

Each of the basic Numeric data types has a color associated with it. Functions that are of the corresponding numeric representation and wire connections that pass that numeric representation are outlined in the corresponding color.

For the Double and Integer numeric representations we have:

Double (64-bit precision floating point) = Orange

I32 (32-bit integer type) = Blue

Creating Numeric Data Type Control and Indicators On the Front Panel

To create a Numeric data type on the Front Panel, we want to make a Numeric Control or Numeric Indicator. When we construct these on the Front Panel, the default Numeric data type is the Double Floating Point Representation. Click [here](#) for how to create a Numeric Control or Indicator.

Creating Numeric Data Type Controls, Indicators, and Constants On the Block Diagram

To create a Numeric data type in the block diagram, we can either create a Numeric Constant from the Functions palette, or we can right-click on a wire and under “Create” we can place either a “Numeric Constant”, “Numeric Control”, or “Numeric Indicator”. As illustrated by Figure 1 below. You will notice that if you create a “Numeric Control” or “Numeric Indicator” on the block diagram, a corresponding control or indicator will appear on the Front Panel design.

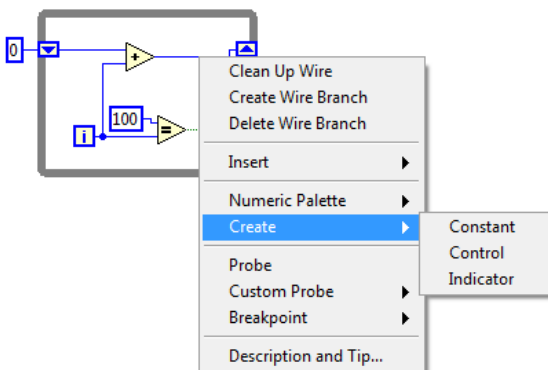
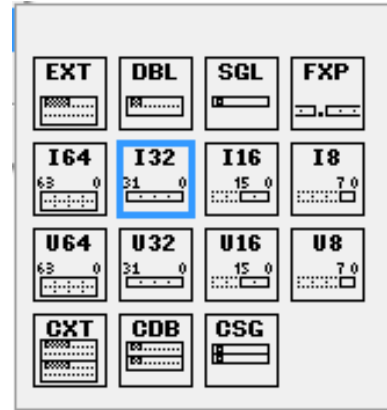


Figure 1

Changing Numeric Data Type Representations

Another property of Numeric data types is the type of representation they use. Common representations are Double Precision Floating Point (Double) and 32-bit Integer (I32). The following is a list of numeric representations that are supported:

Extended Precision Floating Point	Double Precision Floating Point	Single Precision Floating Point	Fixed Point
Quad (I64) – 64-bit Integer	Long (I-32) – 32-bit Integer	Word (I-16) – 16-bit Integer	Byte (I-8) – 8-bit Integer
Unsigned Quad (U64) – 64-bit Unsigned Integer	Unsigned Long (U32) – 32-bit Unsigned Integer	Unsigned Word (U16) – 16-bit Unsigned Integer	Unsigned Byte (U8) – 8-bit Unsigned Integer
Complex Extended (CXT)	Complex Double (CDB)	Complex Single (CSG)	



To change the representation (Figure 2) of a Numeric data type on the block diagram, right-click on the function you want to change the representation for and under “Representation”, select the new representation you wish to change to.

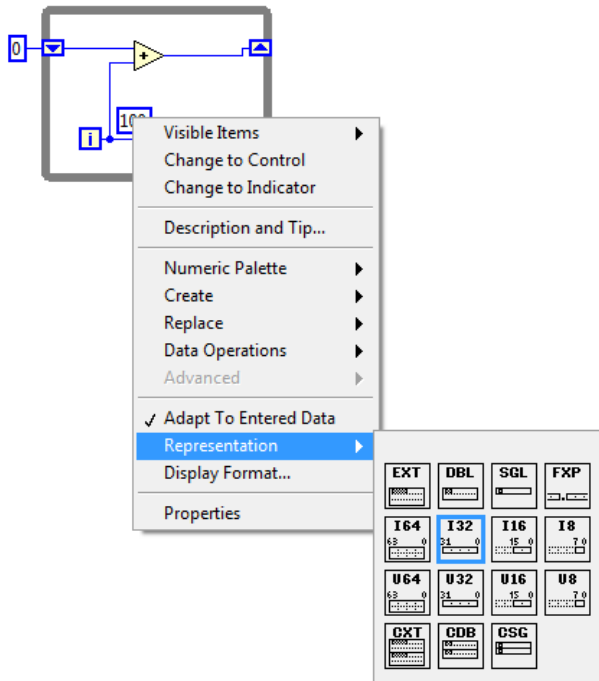


Figure 2

Coercion

Calculations are usually performed using the same data type representations for input and getting an appropriate data type representation for output. If different data type representations are used for inputs, some of the data types may be “coerced” or converted to the same data type representation as the other inputs.

For example, in Figure 1 when we try and add 10 which is an I32 with 1.5 which is a double, LabView coerces 10, which was I32, into a double and outputs a double (a coercion is indicated by a small triangle the color of the data type it was converted to).

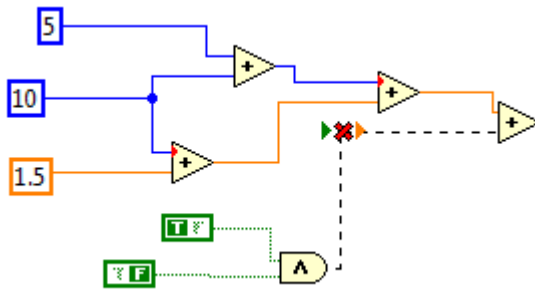


Figure 1

Also notice that when we attempt to connect incorrect numeric representations to a function, LabView indicates this on the diagram with an error symbolized by the red X and the two arrows indicating which data type compatibility was violated. In Figure 1, the error indicates that a Boolean data type was supplied instead of a double.

Finally, a black dashed connection indicates a broken wire. This indicates that the LabView compiler has found an error which needs to be addressed and fixed before execution can proceed.