Intro to LabVIEW™

BLOOMY

Data Types

Discrete

Numeric

Many representations: Floating point, integer, fixed point and more

Boolean

Only 2 states: 'True' or 'False'

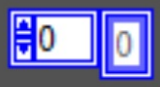
Enum

Integer data that has an associated string with every value

String

ASCII characters (alpha-numeric, symbols)

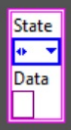
Container



Array

Contains quantity 'N' of single data type

Use when you need an unknown amount of values of a single type (DAQ, Logging, etc.)



Cluster

Can contain any amount of different discrete or container controls

Use when you need to package different data types into a single control/wire (config, UI, shift registers, etc.)

Variant

Generic, can hold any data type

To get data out, you need to know the actual type of the variant value

Use when you need to package different data types into a single control/wire (state machine message packet)

Type Definitions

Pro Tip: A cluster of enum and variant is a flexible state machine message packet

Custom Control: Links neither data nor control appearance to instances in source code

Type Def: Links data but no linking of appearance to all instances in source code

Strict Type Def: Links both data and apperance to all instances in source code

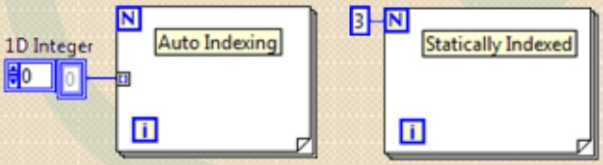
Pro Tip - Always type define any cluster or enum you use throughout your code. You never know when you'll change it

Structures

FOR loop

Runs the contained code a known amount of times

Can be defined as constant 'N' times OR dynamically using auto-indexed array inputs

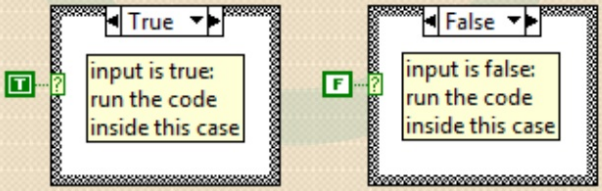


Use for code that needs to run a finite amount of times

CASE structure

Runs the code inside of a single case, dependent on the value at the case selector terminal

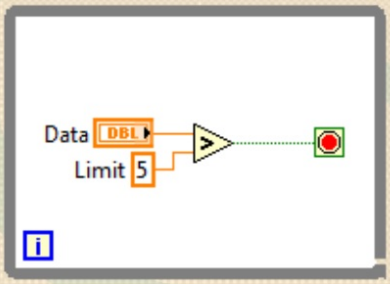
Use for code that you need to make a decision prior to running a set of code ('if, else' statements, state machines)



WHILE loop

Runs contained code until stop condition has been met

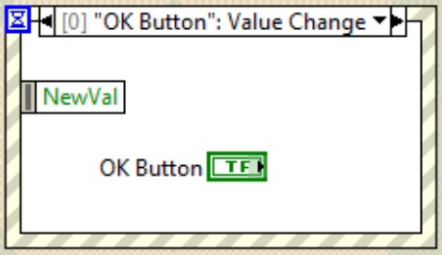
Must run once: stop condition is handled after first iteration



Use for code that will run an unknown amount of times

EVENT structure

Runs the code inside of a single case, dependent on the event that was triggered



You must pre-define which events to handle using the 'Edit Events' dialog

Use for UI interactions (button press, mouse moves, panel close, etc.)

Data Communication

Queues

Every item will be read and executed, even if many items are enqueued in parallel

After data is dequeued, it is removed from the queue

Use when you need to write from many places in parallel, but read from a single place

Notifiers

Broadcasts data to multiple parallel loops

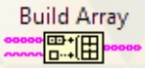
Every time the notifier is written, the last value is overwritten

Use when you need to read from many places in parallel, but write to a single place

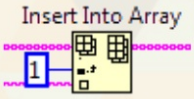
Pro Tip:
Try quick drop instead of the palletes. Press Ctrl+Space and type what you are looking for.

Advanced LabVIEW™

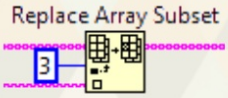
Array Manipulation



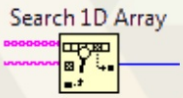
Use this to combine or concatenate scalars or arrays of any dimension



Use this to insert an element or array subset into an array



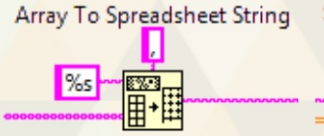
Use this to replace an element into an array

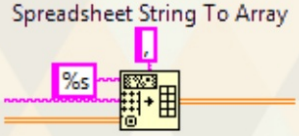


Searches through an array for the FIRST exact match. Wire a '>=0' to the index output to determine if an element value exists or not.

Pro Tip:
Want to get a head start? Use the built-in LabVIEW templates and examples.

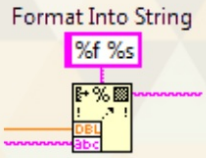
String Manipulation





Use these to convert a 1D or 2D array to a flat string and back, delimited by user-defined symbol. Most used for saving an array of data to a ASCII spreadsheet file (*.txt, *.csv, *.xls)

Flattens any data to a non-human readable string. For human-readable types, use Flatten to XML or Flatten to JSON

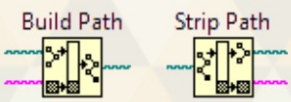


Flattens data inputs to string according to the user-defined 'Format String' input. Use this for creating file headers, etc.

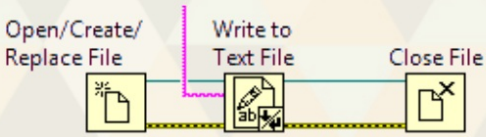
Match on exact string pattern OR regular expression. Commonly chained together, use this for path/string manipulation or dynamic string reformatting

---

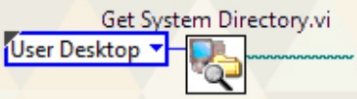
File I/O



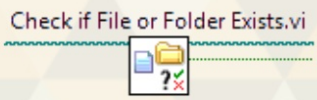
Add or remove string to path input. LabVIEW™ will automatically add or remove the '\'



Use the intrinsic File I/O VIs for greater scalibility, effeciency and for access to the file reference in order to use advanced functions



Use the 'File Constants' palette for run-time access of paths. Typically used in conjunction with build/strip path VIs.

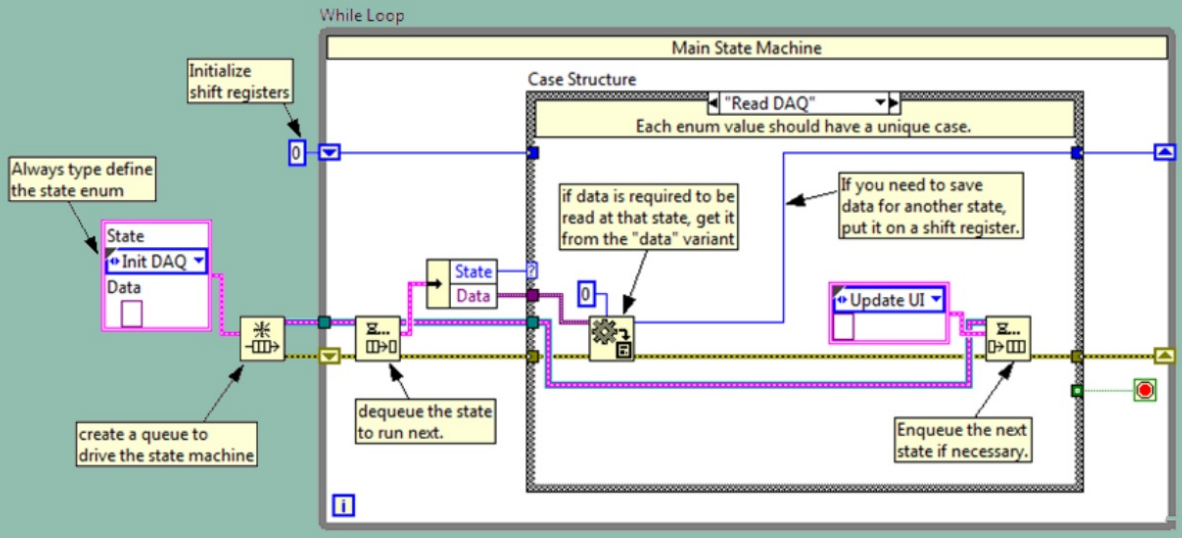


Before reading or overwriting a file, use this VI to check if the file or folder exists to avoid some common file errors

Common Architechures

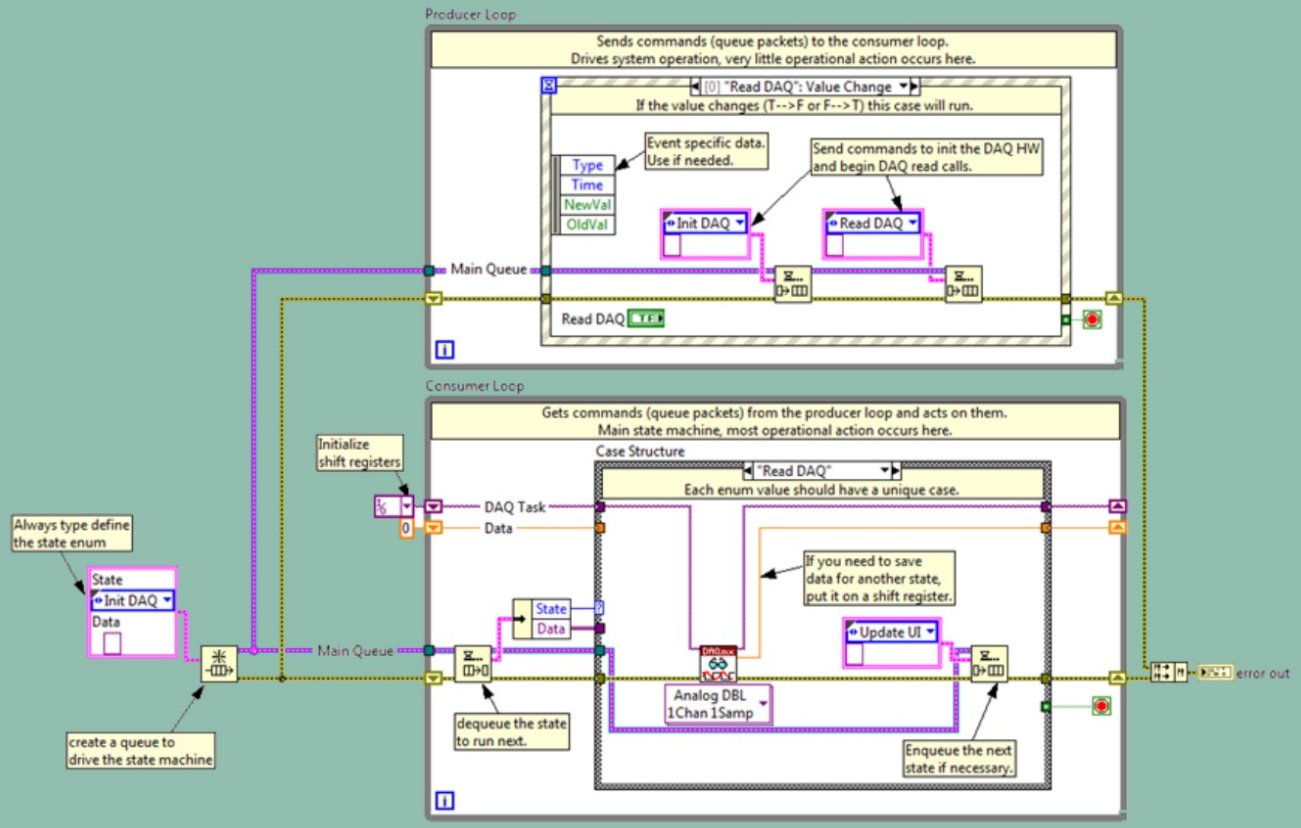
Pro Tip: Always use queues or notifiers to communicate between parallel loops. If used correctly, they can eliminate most race conditions and are a more robust way of communication than variables

Queued State Machine



Producer-Consumer

Pro Tip: The Producer-Consumer model is the most widely used architecture in LabVIEW™ and is a great starting point for a new project



Further Reading :

http://www.ni.com/community/

LabVIEW Quick Reference Card

Bloomy Blog

BLOOMY

The Leader in Automated Test, Data Acquisition, and Control Systems