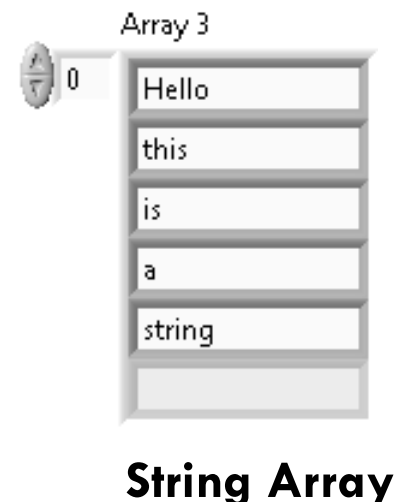
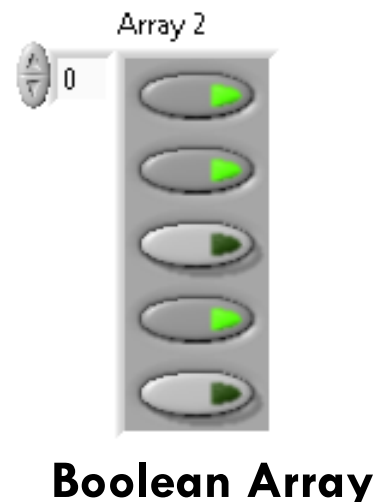
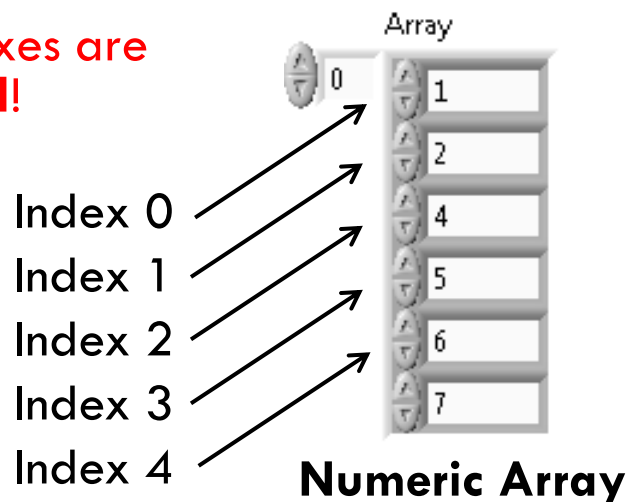


Arrays

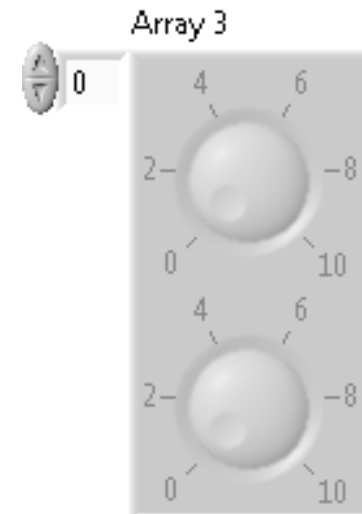
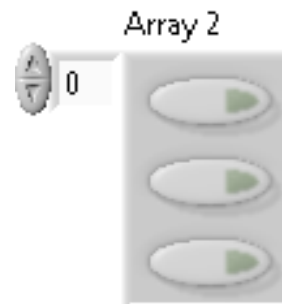
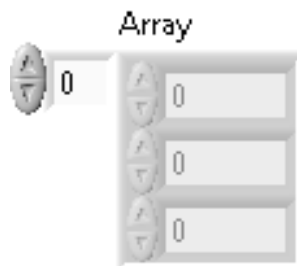
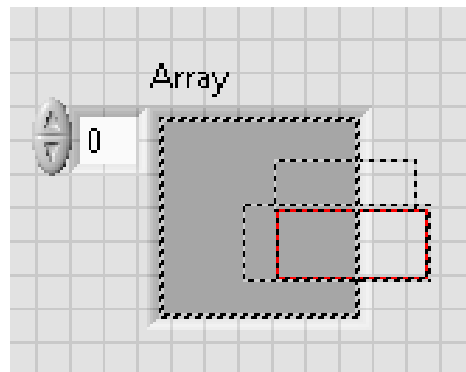
- Combine data points of the **same data type** into one data structure
- One can build an arrays with **numeric, boolean, path, string, waveform** and **cluster data types**

Array indexes are
zero-based!



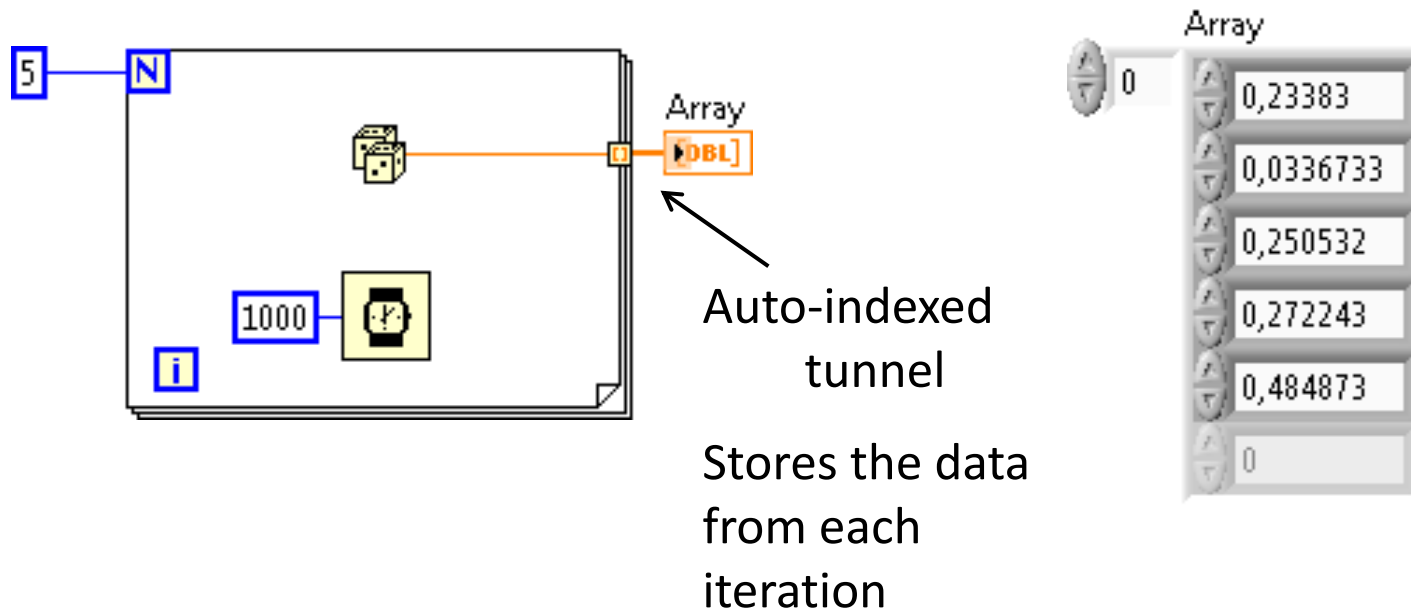
Creating arrays

- Add an **Array Shell**
- Drag a **data object or element** into the array shell



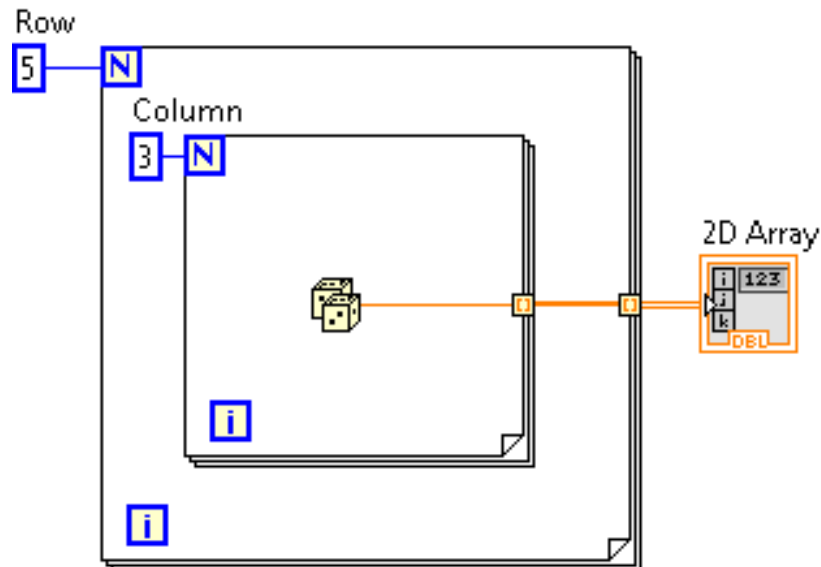
Auto-indexed tunnels

- When you auto-index an **array output tunnel**, the output array receives a new element from every iteration of the loop.



Creating Two-Dimensional Arrays

- Use two For Loops nested one inside the other
- **Outer** For Loop creates the **row** elements
- **Inner** For Loop creates the **column** elements



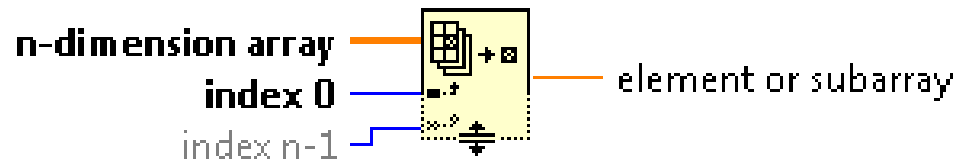
Creates a **5 by 3** 2D array

2D Array

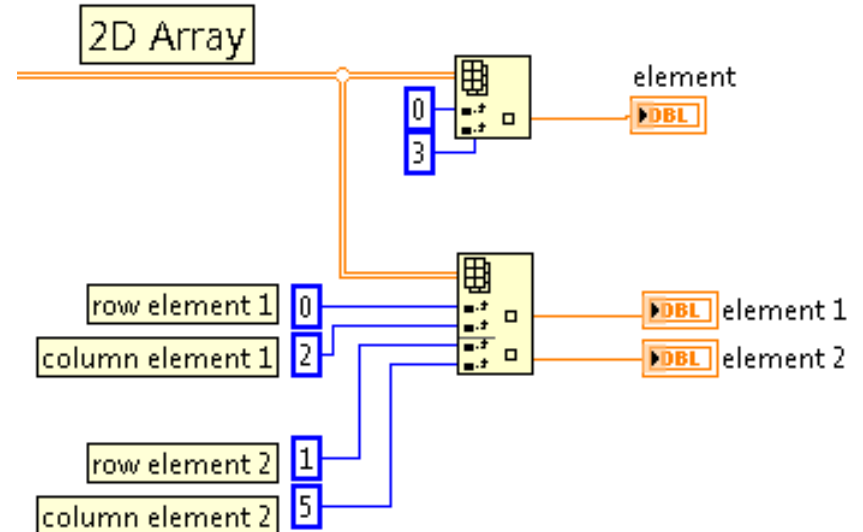
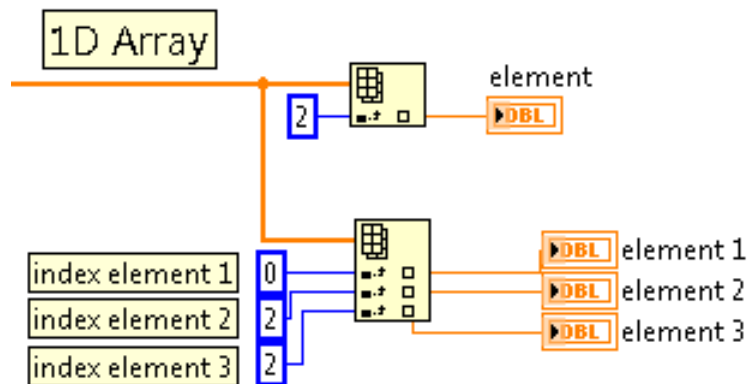
0	0,627429	0,932839	0,698864	0
0	0,393791	0,924806	0,384784	0
	0,869544	0,246031	0,636578	0
	0,092679	0,252591	0,120121	0
	0,908866	0,133016	0,717746	0
0	0	0	0	0

Index Array

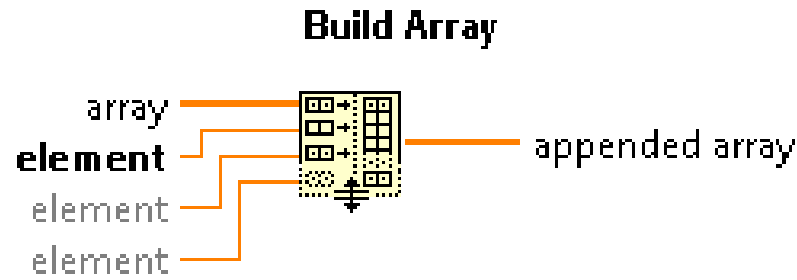
Index Array



Returns the **element or subarray** of **n-dimension array** at **index**.

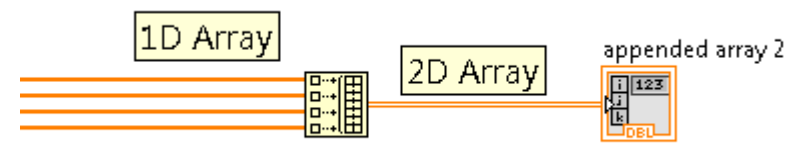
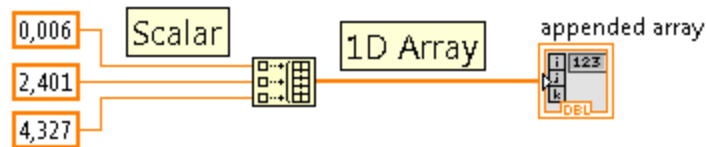


Build array

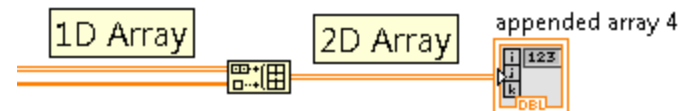
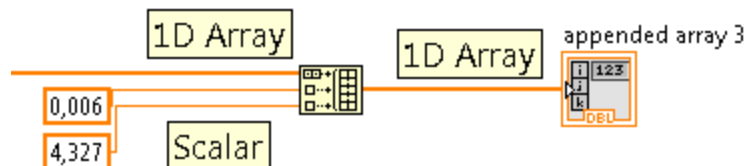


Concatenates multiple arrays or appends elements to an n-dimensional array.

Concatenate examples

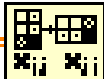


Append examples

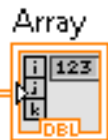
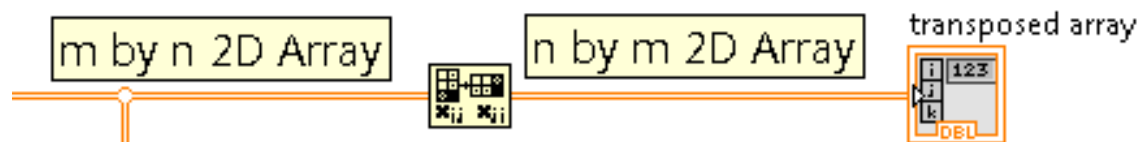


Transpose 2D Array

Transpose 2D Array

2D array  transposed array

Rearranges the elements of **2D array** such that **2D array**[i,j] becomes **transposed array**[j,i].



Array

0	0,47999	0,40851	0,84910	0
0	0,72783	0,05409	0,35303	0
	0,83583	0,64280	0,97105	0
	0,63069	0,68472	0,10432	0
	0,33184	0,14612	0,39571	0
	0	0	0	0

transposed array

0	0,47999	0,72783	0,83583	0,63069	0,33184	0
0	0,40851	0,05409	0,64280	0,68472	0,14612	0
	0,84910	0,35303	0,97105	0,10432	0,39571	0
	0	0	0	0	0	0