

## Appendix IV – NSLJ and NSLC Differences

---

There are several small differences between the NSLC and NSLJ implementations.

### A.IV.1 Ports

One such difference is the way in which the input and output ports are implemented on a module. In the NSLJ version, output ports allocate memory whereas input ports do not. In the NSLC version both input and output ports allocate memory.

### A.IV.2 Read/Write Script Access

Another difference, is the fact that the NSLJ system actually implements the “nslSetAccess” methods and thus variables that do not have a ‘W’ access associated with them are not manipulatable from the scripting environment nor from the other modules. The current NSLC version does not implement “nslSetAccess,” thus providing a default ‘W’ access for all variables.

### A.IV.3 Frames and Modules

Another difference is the fact that in NSLJ all **NslOutFrames** and **NslInFrames** automatically create a **NslOutModule** or **NslInModule** respectively. This is due to the fact that the modules are the objects that actually get scheduled by the scheduler and not the frames.

### A.IV.4 NslBoolean

Another difference is the NslBoolean class. NSLJ allows boolean arrays from dimension 0 to 4 and provides a number of methods to manipulate and compare boolean arrays. NSLC treats the NslBoolean class as NslInt.

### A.IV.5 Methods

One important thing to note about the NSLJ mathematical methods is that since it is not possible to provide every combination of parameters when parameters can be one of six different types, it was decided to implement the most “logical” combinations. Typically this means that if the method takes more than one parameter, the parameters and result should be of all the same base type (**int**, **float**, **double**, **NslInt**, **NslFloat**, **NslDouble**). NSLC implements the different combinations as templates.

