

Type and concurrency

Access to variables is always serialized so no explicit synchronization is needed by the user as long as a variable of the appropriate type is used for the situation. Each variables type will determine the concurrency semantics in a given situation. For example, if a global variable is used to store a string that is read and written by two different rule sets the reads and writes are guaranteed to be atomic. However the current value will be determined by the last rule set instance to write to the variable.

If a rule set variable is used to store transient information in a long running rule chain the stored value will most likely no longer be assigned to the variable by the time it is retrieved. More than likely it will have been overwritten by later executions of the rule set. If however a temporary variable were used the value will still be correct when it is retrieved as each execution will have its own instance of the variable.

The following table summarizes the possible variable types for each usage:

Usage	Variable Type		
	Global	Rule Set	Temporary
Isolating external references	Yes	Maybe	No
Simplifying complex event property references	No	No	Yes
Persisting processing or execution state	Maybe	Maybe	Yes