
0.1 Inclusion of Foreign Language Code

Foreign Language Code - as it is referred by this section - is functionality that is included into SystemVerilog using the DirectC Interface. As a result, all statements of this section do apply only to code included using this interface; code included by using other interfaces (e.g. PLI, VPI) is outside the scope of this section. Due to the nature of the DirectC Interface, most Foreign Language Code will usually be created from C or C++ source code, although nothing precludes the creation of appropriate object code from other languages. This section adheres to this rule, it's content is independent from the actual language used.

In general, Foreign Language Code may be provided in form of object code (compiled for the actual platform) or in form of source code. The capability to include Foreign Language Code in object code form is a mandatory feature that must be supported by all simulators according to the guidelines in this section.¹ The capability to include Foreign Language Code in source code form is optional, but must follow the guidelines in this section, when it is supported by a simulator. The inclusion of object and source code is assumed to be orthogonal and must not be dependent on each other. Any interferences between both inclusion capabilities should be avoided.

This section defines

- how to specify the location of the corresponding files within the file system
- how to specify the files to be loaded (in case of object code) or
- how to specify the files to be processed (in case of source code)
- in which form object code has to be provided (as a shared library or archive)
- how to specify compilation information required for processing source code

It does not define

- how to organize the corresponding files within the system (a certain directory structure)
- how object code files are loaded
- how source code files are processed and finally included in a simulation (besides the requirement that this must be fully transparent to the user)

All pathnames specified within this section are intended to be location independent, which is accomplished by the environment variable "SV_ROOT" and the corresponding switch "-sv_root". Both can receive a single directory pathname as value, which is then prepended to any relative pathname that has been specified.

It is a general rule defined for all settings in this section, that a switch value overrides the value of the corresponding environment variable. Therefore, the value of the -sv_root switch overrides the value of the SV_ROOT environment variable.

Furthermore it possible to completely turn off the usage of all environment variables defined within this section by specifying the switch '-sv_no_env' (without any value). Any option/switch processed after processing this switch must ignore any related environment variable settings in existence.

¹.Object code inclusion is mandatory