# Spice Analyzer

## High Performance Spice Analysis

## Highlights

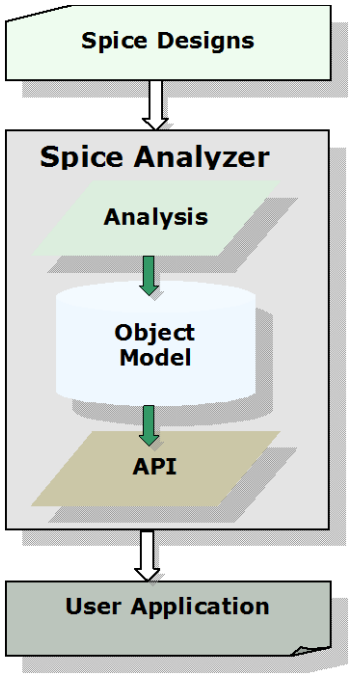
- High performance Spice analysis
- Advanced semantic analysis that performs validity checks on parameters, resolves models including mapping of binned models, and more
- Well-defined, complete set of API functions that allow access, modification, and creation of all possible constructs
- API functions for static expression and function evaluation
- Support for incomplete designs through black box analysis
- Flattening utility to flatten, partially flatten, and un-flatten instantiation hierarchy
- Complete support for parameters and their evaluation in global and local scope
- Support for exact decompilation of comments in the design file
- Browser utility for easy debugging by traversing the object model
- User controllable parasitic extraction utility that removes parasitic elements

Interra's Spice Analyzer addresses the need of EDA tool developers who need to add support in their products for a comprehensive analysis of Spice syntax and semantics. Targeted as a customizable front-end for applications, such as Simulation, DFM, Timing Closure, and Power Analysis, the Spice Analyzer is compliant with industry standard simulators and other tools.

With a common architecture and object model to support different Spice variants, the analyzer offers several advantages, such as cross platform decompilation and easy integration with different tools.

The Spice Analyzer's easy-to-integrate C++ API is intuitive and comprehensive enabling EDA tool developers to analyze as well as access designs for information, modify designs, evaluate functions, perform elaboration, and more.

Backed by Interra's field proven expertise in developing language analyzers, the Spice Analyzer offers a best value solution and reduces time-to-market standard EDA products.




## Key Advantages

- Extensible architecture to support various Spice variants
- Common object model to ensure easy integration
- Comprehensive coverage of HSPICE constructs. Supports HSpice 2009.09
- Complete support for Spectre
- Comprehensive validation of syntax and semantics
- Is backed by Interra's field-proven expertise in developing analyzers

# The Spice Analyzer Features

## Comprehensive Support

Comprehensive support for analysis of Spice netlists, covering all features and constructs.

## C++ Procedural Interface (API)

The Analyzer provides C++ procedural interface to integrate the Analyzer with C++ and C applications. The API is comprehensive, covering functions, such as flattening and parameter evaluation. The API functions also enable a lot of customization, such as creating and attaching user-defined attributes to objects and specifying analysis options.

The intuitive API function names facilitate a short learning curve and provide better understanding on how to use the API functions.

## Semantic Checking

The Spice Analyzer provides extensive semantic checking of parameters. The semantic checking includes elaboration of units, mapping of model references, resolving hierarchical references to nodes and devices, checking for loops in parameters and voltage sources, and correct identification of parameters even when LHS is not specified.

## Dynamic and Extensible Object Model

The in-memory, C++ object model, created after analysis, is dynamic and extensible. You can use API functions to add/remove elements, rename elements, add/remove parameters, and connect/disconnect nodes. You can modify and extend any object. Further, the analyzer provides factory classes to create and add new objects to the object model. For example, a new user defined device class object may be created and added to an existing subcircuit. The extended object model can easily be decomplied as the modified netlist!

## Expression and Function Evaluation

You can use the API functions to evaluate static expression and functions. An expression having constant values can be evaluated to static constant value. In addition, you can evaluate expressions that involve in-built functions and parameters. You can also determine the expression size in static functions. In built functions are identified and evaluated statically, such as sin and exp.

## Decompilation

Using the API functions, you can easily decompile any object in the object model. A complete dump of the input Spice netlist can be obtained at any hierarchy level. In addition, the decompilation API functions automatically decompile the attached comments with the construct.

## Exact Decompilation

You can decompile the Spice netlist ‘as is’, such that the decompiled output preserves the order of the statements, the order within the statements, and the file structure. The spaces, tabs, and new lines are maintained as well!

## Elaboration

API functions are available for partial elaboration of the object model. You can select a particular hierarchy path and call API functions to resolve references based on this path. Further, any subsequent access to the parameters of device instances in this path would return the overridden values obtained during elaboration of this path.

## Flattening

Your applications can call API functions for flattening the instantiation hierarchy and achieve a flattened structure. During flattening, the Analyzer does a proper mapping of the terminals and also promotes parameters.

## Black Box Analysis

The Spice Analyzer allows parsing and the creation of an object model from a design that has missing subcircuit references or models. The models or subcircuits are treated as equivalent devices and all their instances are thus treated as an equivalent device statement. For example, a subcircuit or model can be treated as an equivalent resistor, thus making all instances of the subcircuit behave as either resistor or instance.

For all missing subcircuits or models, a dummy container is created with a list of inputs and output ports inferred through corresponding instantiations.

## Parameter Handling

Parameters are effectively handled and resolved through instance hierarchy in both local and global scope. You can use API functions to evaluate RHS having expressions and functions. Further, functions are available to access, modify, and add parameters. You can even access list of referenced parameters in a subcircuit, identify loops in parameter definitions, and validate parameter values based on a given valid value set!

## Customizable Error Handling

The API functions enable applications to customize the messages reported by the Spice Analyzer to suit application-specific needs. Applications can use the API functions to suppress error messages and warning messages or change the severity of messages.

**Interra Systems, Inc.**
1601 S. De Anza Boulevard, Suite 212, Cupertino, CA 95014
Phone: +1 408 579 2000, Fax: +1 408 579 2050
Email: info@interradesign.com Web: www.interradesign.com