

# TOUCHSTONE 2.0 PARSER DEVELOPMENT REQUIREMENTS

May 6, 2009

The following is a general description of the requirements for a text-based computer program to parse and check the syntax of Touchstone 2.0 data files.

## a) LANGUAGE

The code shall be written in ANSI C. C++ object programming features are not used.

## b) GENERAL REQUIREMENTS

- 1) All keywords, subparameters, behaviors and relationships defined in the Touchstone 2.0 specification shall be checked as used in parsed files for compliance to the specification. The key features and keywords are:

- [Version]
- # (the options line)
- [Number of Ports]
- [Two-Port Data Order]
- [Number of Frequencies]
- [Number of Noise Frequencies]
- [Reference]
- [Matrix Format]
- [Mixed-Mode Order]
- [Begin Information]/[End Information]
- [Network Data]
- [Noise Data]
- [End]
- Comments
- Single-ended network parameter data
- Mixed-mode network parameter data

- 2) The order of the keywords, option line and network data in the input files relative to each other shall be checked for consistency with the requirements of the Touchstone 2.0 specification.
- 3) Violations of explicit rules in the Touchstone 2.0 specification present in the file shall be reported to the user as errors. These can include but are not limited to missing or misspelled keywords, incorrectly ordered keywords or features, or inconsistencies between arguments for related keywords (e.g., [Number of Frequencies] and the data actually present under [Network Data]). Potential errors in the data or issues of data interpretation shall be reported as warnings (e.g., extremely large or extremely small data values).
- 4) In addition, warning and error messages shall be assigned unique numbers where possible (see below).

- 5) No parsing of original Touchstone (also known as Touchstone 1.0) files is assumed or required by this specification.

### **c) USER INTERFACE**

The parser is to be accessible through a command line interface, using a command to be determined. The parser shall accept a single Touchstone file name as an argument.

Output shall consist of warnings and/or errors, listing the type of error and the line number of the original file where the problem occurs. A summary of errors and warnings shall appear at the end of the output listing.

The parser shall also divide the input into tokens and other database structures to permit integration of the parser with simulation and/or laboratory tool software.

### **d) OPERATING SYSTEMS**

The source code shall be easily compiled for Microsoft Windows\*, Linux, and major versions of UNIX. Use of separate source code sets for compilation under different operating systems is acceptable, but differences between OS-specific source code sets should be minimized.

### **e) QUALITY ASSURANCE (QA) SUITE**

A set of test cases, consisting of text files in Touchstone 2.0 format, shall be provided to prove the functionality of the parser when analyzing Touchstone 2.0 files. A script, batch file or other automatic means of running the test cases shall be provided along with the source code distribution. Where applicable, the test cases shall include examples of error handling and messages that result.

### **f) NUMBERED ERRORS AND WARNINGS**

All errors and warnings issued from the Touchstone 2.0 parser shall be identified by a unique number. This number shall appear before the error/warning text itself in the parser output. Where possible, the line number where the error occurs in the parsed file shall be reported as part of the notification to the user. Note that not every error or warning can be assigned a line number (e.g., a missing keyword will have no line number).

### **g) REFERENCES**

Please refer to the Touchstone 2.0 specification for specific rules and details. The document can be found at:

[http://www.eda.org/ibis/touchstone\\_ver2.0/](http://www.eda.org/ibis/touchstone_ver2.0/)

\* Other brands and names are the property of their respective owners