

GenCAM Syntax Checker

Abstract

A conformance test module (CTM) for the GenCAM file format is being developed by National Institute of Standards and Technology (NIST) to support an industry effort to develop the GenCAM standard. GenCAM is a product data exchange standard for information used in the manufacturing of printed boards and printed board assemblies. This initial release of the CTM provides level 0 testing for the file format. This limited test only checks the syntax of the file. Past experience convinced the GenCAM committee that a CTM is a key requirement for the success for the standard. This prompted the request from industry to NIST to develop the CTM. NIST will release additional levels of testing for the CTM as they are developed.

Introduction

The Institute for Interconnecting and Packaging Electronic Circuits (IPC) is a trade association representing over 2300 companies in the electronic interconnection industry worldwide. This IPC works with industry volunteers to develop standards to support the manufacture of printed board and printed board assembly products. The GenCAM standard is a new standard that was developed by the IPC Data Transfer Solutions (DTS) committee. GenCAM is a standard data file format that is used to describe printed board and printed board assembly products with details sufficient for tooling, manufacturing, assembly, inspection, and testing requirements. These formats may be used for transmitting information between a printed board designer and a manufacturing or assembly facility. The files are also useful when the manufacturing cycle includes automated processes and numerical control machines.

When the DTS committee established the scope and requirements for GenCAM they identified some requirements that were considered essential for the success of GenCAM. Among these requirements was the need for a conformance test module (CTM) for validating vendor implementations of tools that read and write GenCAM files. One aspect of the CTM is a software application that reads in a GenCAM file and verifies that the syntax and semantics of the file conform to the definition of a GenCAM file as defined in the GenCAM specification. The second part of the GenCAM CTM will require the creation of a set of GenCAM "golden board" files that will test each feature of the GenCAM specification. The first parts of the CTM will test the ability of a tool to write a valid GenCAM file and the second part will test the ability to read a GenCAM file.

The Scope of the CTM Tests

The full CTM will include both syntax and semantic testing. The syntax checker only checks the punctuation of the GenCAM file. That is, it only checks that the statements in the file use the correct tokens, such as keywords, numbers, and strings, and that tokens are used correctly within each of the statements within the file. The semantic checking of a GenCAM file will examine the content of each statement and the relationships between statements in the file. The semantic tests will be designed to verify specific definitions found in the GenCAM specification.

The complexity of the interactions between the GenCAM objects, as defined in the GenCAM specification, makes the creation of a tool that can check for all possible semantic errors cost prohibitive. The CTM documentation will include a description of all tests that are made on a GenCAM file. This does not mean that a GenCAM file that passes the CTM tests is necessarily a valid GenCAM file or a functional product design. Passing the CTM tests only means that no errors of the type the CTM was designed to locate were detected.

The CTM will structure the semantic checks into layers. Higher layer numbers will correspond to tests that are more difficult to perform. The level 0 release of the CTM contains only a simple syntax checker (no semantic checking is included in this release). The level 1 semantic checking will verify that parameters in keyword statements are of the correct type. The level 2 test will verify the content of each GenCAM section. If a keyword statement is located in a scope which is not defined in the BNF, it will flag an error.

This release of the CTM and this document only describe and test the use of the CTM release level 0.

System Specific Installation Instructions



No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

<p>Linux and Unix platforms</p>	<p>How to Install the Syntax Checker</p> <p>The software necessary to run the CTM syntax checker is found in syntaxchecker.tgz. This file contains the Pmw package, the GenCAM package, some test data, an install.py script and the gcsc program (a graphic user interface to the GenCAM syntax checker).</p> <p>This release of the CTM requires that Python 1.5.1 with the Tkinter module be installed on the system. Most Linux distributions come with the Python and Tcl/Tk preinstalled so those packages are not included with the syntaxchecker.tgz distribution. An RPM package of Python 1.5.1 is available if this version of Python isn't already installed on your system.</p> <p>For other Unix platforms see the Python web site for instructions for installing Python on a specific system.</p> <p>After verifying that Python 1.5.1 is installed, untar the CTM distribution. Next, as root, run the install.py script. The script will install the packages Pmw and GenCAM in the site-python directory and install an executable named gcsc in /usr/local/bin. The data directory contains examples of GenCAM files. The data directory is not installed by the install.py.</p> <p>How to Use the Syntax Checker</p> <p>Execute the gcsc program to start the syntax checker. Under the File menu select the Open command. Select a GenCAM file (look in the data directory for several examples) and press "Ok". The Parser will read the GenCAM file and report any errors found in an error window.</p>
<p>Microsoft Windows 95</p>	<p>How to Install the Syntax Checker</p> <p>All the software required to install and run the CTM release level 0 on Windows 95 is included in the syntaxchecker.zip file. The current release has only been tested on Windows 95. It will probably work unaltered on Windows NT and Windows 98. The graphic user interface for the tool will not work on Windows 3.1. (The syntax checker could be run as a command line program at a DOS prompt. Instructions on doing so can be supplied on request.)</p> <p>To install the CTM on Windows unzip the syntaxchecker.zip file. This file contains the install scripts for Python and Tcl/Tk and an install script for the CTM related software (only the syntax checker at this time.) Python requires less than 6M of disk space to install. Tcl/Tk require about 6M of disk space to install. Use the default install directories as prompted by the install programs if possible. While it should work, the SyntaxChecker has not been tested with the software packages installed in other directories. If either Python 1.5.1 or Tcl/Tk 8.0 are already installed on the system, then installation of those tools is not required.</p> <p>If Tcl/Tk is not available on your system, then run tcl803.exe. This program installs Tcl/Tk. A dialog box will ask what kind of install to use. Select the "full install" option. The Tcl/Tk installation process requires rebooting after the install is finished.</p>

If Python is not available on your system then, run the **pyth151.exe** program to install Python. **Reboot once again after this install script has finished.** The **reboots after installing Python and Tcl/Tk** are required or the syntax checker will not work.

Once both Python and Tcl/Tk are installed, the final step is to run the script **installCTM.py** (the snake icon). This will install several Python packages into the Python search path. One package is a general purpose Python graphics package called Pmw and the other is the GenCAM specific Python package, called GenCAM.

The syntaxchecker.zip distribution includes a program called SyntaxChecker. Executing this program starts up the graphic user interface for the syntax checker application. The distribution also contains a data directory with several example GenCAM files. (See description below.)

How to Use the Syntax Checker

Double click on SyntaxChecker to start the applications. A window will appear on the screen with a menu across the top. Under the File menu select the Open command. Select a GenCAM file (look in the data directory for several examples) and press "Ok". The application will open two scrollable text boxes under the menu. The top box contains the source code read from the GenCAM file. The bottom box will display any error messages generated by the syntax parser.

The Examples in the Data Directory

The data directory included in the distribution of the syntax checker includes four example data files: 2511final3.gcm, typical.gcm, failure1.gcm, and failure2.gcm. The 2511final3.gcm file is a complete collection of all the examples from the October 1998 release of the GenCAM specification. Some of the examples are not located in a GenCAM "section". These miscellaneous examples were grouped in a \$PREAMBLE section so that the syntax checker would not generate a syntax error when the example statements were parsed. The \$PREAMBLE section was created simply to support the testing of all the examples in the document for syntactic correctness. Since the GenCAM specification has no \$PREAMBLE section the 2511final3.gcm file is obviously not a semantically compliant GenCAM file.

The typical.gcm file contains just the examples from section 4.8 of the GenCAM specification. Section 4.8 describes all of the GenCAM file "sections", so the typical.gcm file has one example for each type of GenCAM file "section". The typical.gcm file may resemble what a typical GenCAM file might look like, but the file has only been tested for generic correctness of each statement's syntax. No checks are made to see if statements contains the correct number and type of parameters or whether statements are properly located within sections or within the scope of statements within a section.

The failure1.gcm and failure2.gcm files are copies of the typical.gcm file, but with several simple syntax errors introduced.

Information on Tools Used to Develop the CTM

The CTM was developed using the Linux operating system and the Python language. The Linux OS is an open source clone of Unix. Python is an open source, platform independent, object oriented scripting language. The source code and executables for both are available at many sites on the Internet, or pre-packaged on CDROM by a number of software vendors.

The Linux OS is a non-proprietary, open system which is has been ported to all of the major microprocessors (i.e. 8x86, 68xxx, MIPS, SPARC, PowerPC, Alpha, and ARM). Two aspects of Linux make it a good choice for developing software for the CTM. The complete source for the OS is available for peer review, so the chances of hidden bugs are minimized. The open source process for developing software has been proven to be a cost effective

and reliable means of producing robust and clean software. Second, the developers of Linux have integrated interfaces for many industry standards for software, such as POSIX, X Windows, and all of the Internet protocols.

The Python language is an extensible object-oriented language which is written in C. The language organizes software into hierarchical packages of modules. Modules can be written in the Python language or in foreign language such as C or C++. Python's dictionary-centric, object-oriented implementation makes it a good match for the string-based object naming nature of GenCAM. As an embeddable and extendable language, the GenCAM CTM could be used by application developers to directly embed a standard GenCAM reader and writer in an existing application. Python's abstract C API could then be used to manipulate data and share data between the application and the GenCAM parser. The ability to use Python as a high-level report generation language could be used by industry to develop standard GenCAM reports.

The GenCAM Syntax Checker Source Code

The GenCAM syntax checker is split into two modules that are located in the GenCAM package. The `SyntaxParser` module provides classes for reading in a GenCAM file and generates a list of errors and a data structure that holds the parsed data. The `SyntaxOnlyChecker` module provides a graphical user interface front end to the `SyntaxParser` module. The GenCAM and Pmw packages are installed in the `site-python` directory in the Python path on Linux and UNIX and in `Python` directory in the Python path on Windows.

On Linux and UNIX the `gscs` executable script, which starts up the graphical user interface for the syntax checker, is placed in `/usr/local/bin` by the installation script. For Windows, an application called `SyntaxChecker` is provided in the zip file. This program can be copied to any suitable location on the Windows system.

External Tools Used by the CTM

The installation of the CTM requires that the Python language interpreter, Tcl/Tk, and Python megawidget package be installed on the system. References on each of these tools are available at:

- [Python Language References](#)
- [Tcl/Tk References](#)
- [Python PMW package](#)

Michael McLay

Last modified: Mon Oct 5 14:41:44 EDT 1998