



Command-Line Interface

The command-line interface (CLI) tools allow scripting of various functions performed by the WAE Design GUI, such as import, export, initialization, simulation and optimization. CLI tools can, in some cases, provide additional options that are not available from the GUI, such as network interface tools. This chapter describes the CLI options file and logging levels.

CLI Options File

Options for any tool can be invoked by specifying these options at the CLI, or by specifying the options in an `options.txt` file. By default, each CLI tool looks for this options file in the `$CARIDEN_HOME/etc` directory. A different options file can be specified when calling any of the tools by using the `-options-file` option.



Note

Options specified at the command-line take precedence over options specified in the options file.

The Options file format consists of zero or more lines of the following form.

```
<tool-name>:<option>=<option-value>
```

Here `<tool-name>` is the tool name as specified in the table above, `<option>` and `<option-value>` are an option and its value, specific to that tool.

Lines starting with the character `#` are treated as comments and ignored.

If a filename is specified as a parameter to an option in `options.txt` using a relative path or without a path, the path will be relative to the directory in which the tool is called, and not the directory containing the `options.txt` file.

When tools are called through the GUI, the default options displayed in the dialog correspond to the options file settings. Note that changes in the options dialog entries will however remain in effect for the remainder of the session, or until changed.

Options to each command-line tool are available by calling the tool with the `-help` option.

Logging Levels

Many commands have an option (`-verbosity`) that controls the level of logging.

Table 9-1 Log Levels

Level	Type	Description
0	Fatal	Premature termination of a CLI tool or GUI process under circumstances beyond our control, so output might be corrupt. Minimal effort to handle it elegantly. Example: disk full, cannot allocate memory.
1-10	Error	An error occurred so that the results of the tool might be wrong, but there will still be output. Probably premature termination. Example: couldn't log in to the seed router for snmp_find_igp_db, bad import format, bad .pln file format.
11-20	Warning	Use of deprecated API, poor use of API, unexpected or undesirable situations. While having more information to continue is preferable, a best guess is made. Examples: Replacing unsupported characters on import, updating SRLG table format on import, unequal capacities on import
21-30	Notice	Results of a tool that can be used to make a decision about what to do next. Examples: which routers could not find communities.
31-40	Info	Useful information to further assist users. Can be per object. Timing information for long processes. Examples: Metric Optimization completed in 45 seconds, Tried to get community for node A, succeeded
41-50	Debug	Detailed information of flow through system to help trace back an error or warning. SNMP queries: router needed to be retried.
51-60	Trace	Detailed information of flow through system to help trace back an error or warning. Examples: individual SNMP queries, Full Mosek output for demand deduction, metric optimization