

SCALD User Interface ValidCOMPERR

Revision 9-18-85

(Program Release ValidCOMPERR 1.0.0)

07 Nov 85 Tivonne Ha

Copyright 1985

Valid Logic Systems Incorporated

This document contains confidential proprietary information which is not to be disclosed to unauthorized persons without the written consent of an officer of Valid Logic Systems Incorporated.

The copyright notice appearing above is included to provide statutory protection in the event of unauthorized or unintentional public disclosure.

Copy Number ____

1.0 Introduction

The ValidCOMPILER compiles drawings one page at a time, producing a "list" file and an "expansion" file for each page. This occurs for all drawings except plumbing bodies, whether or not the drawing is parameterized. Context sensitive drawings have their pages compiled (one at a time) for particular contexts so that one "expansion" file and one "list" file are produced per context per page.

Given this design philosophy, ValidCOMPERR allows the user to view the error listings (if any) of the modules invoked in his design.

ValidCOMPERR is run after the ValidCOMPILER is run for a design.

ValidCOMPERR reads the following files:

- schema - The organization of context sensitivities important to separate compilation. This file is fully described in the Separate Compilation Schema ERS. This file is stored under the drawing directory.
- infile - The compiler directives file known as compiler.cmd. This is a logical file.
- expansion - The results of the compilation. This file is stored under the drawing directory.
- list - Serves the same function as the 7.0 cmlst file. This file is stored under the drawing directory.
- lnklst - The error listing of the linker. This file is stored under the root drawing directory.

ValidCOMPERR produces the following result files. (More detailed descriptions follow in other sections.)

- monitor - Serves the same function as the 7.0 monitor file. This is a logical file.
- OutGetErr - A collection of all error listings (if any) of a given design. This is a logical file.

2.0 Nomenclature for ValidCOMPERR

The following definitions are used in describing ValidCOMPERR. (These words may occasionally also be used in their more usual sense.)

schema: A precise description of the context sensitivities of a drawing whose values are instance dependent and thus can require that different compiler results be generated for different instances. This set of sensitivities is taken from the body drawing (version 1). A drawing with a size parameter would have this fact noted in its schema. A drawing with no context sensitivity would have a null schema.

context: A "valued" schema for a drawing. By specifying a value for each context sensitivity of a drawing, we get something that can be compiled. A drawing with a size parameter and no other context sensitivities could be compiled for contexts SIZE=1, SIZE=2, etc. It would have one context for each different value of SIZE that was actually used. A drawing with no context sensitivity would have one context that is described by a null list or empty set.

module:

1. A separately compilable subset of a hierarchical schematic diagram. A module can have an associated expansion file. In previous releases of the Compiler, this was a non-parameterized drawing. In the 1.0 release of the ValidCOMPILER, this will be a page of a drawing with a particular context specified. "Drawing alu page 1 for SIZE=2" could be a module in release 1.0.
2. A drawing and context specification (release 1.0 and later only). This would imply all of the pages of the drawing that apply to that context. "Drawing alu for SIZE=2" specifies a module that may contain more than one page and thus have more than one associated expansion file.

design: A set of drawings containing a root drawing and all of the modules under it.

3.0 File Names

The schema, expansion, and list files are stored under the directory of the drawing they represent. The schema file is named "schema". There is only one schema file per drawing. There is only one lnkfst file per design. There can be more than one expansion and list file per page (one per page per context). Expand file names are formed as follows:

```
<extension>_exp.<version>.<page>.<context>
```

where <extension>, <version>, and <page> are identifier, natural number and natural number (as in other file names within the directory) -- and <context> is a short string (often just a number) indicating size, version, etc... List file names are the same except that _lst is substituted for _exp. Lnkfst file name is formed as follows:

```
<comptype>_lli.<version>.<context>
```

Examples:

```
logic_exp.1.1.1  logic_exp.1.1.2  time_exp.1.1.2a  time_exp.1.1.2b
logic_lst.1.1.0  sim_lst.1.1.0    time_lst.2.5.2a  time_lst.1.1.1
logic_lli.1.1.0  sim_lli.1.1.1    time_lli.2.0a
```

4.0 Details of COMPERR Command

In the following, letter case is not important for command arguments. That is, the argument may be in upper case, lower case, or mixed case. If the argument contains "special" characters (characters other than letters and digits), it should be placed in quotes. Command names should be in lower case. Note that the program ValidCOMPERR is supported only under UNIX.

The command line arguments for <drawing> and <compile type> in the following command replace directives in the directives file. If a command argument appears for the ROOT_DRAWING or COMPILE directive, the corresponding directive is ignored in the directives file. If both the <compile type> and the COMPILE directive are not specified, LOGIC is assumed. <context> is the context specification for which the drawing is to be shown. An example of a <context> specification is: ["<id1>=<value1> <id2>=<value2>..."]. If the context <id> is not specified, the context <value> parsed is associated with the context SIZE. If <context> is not specified, it is default to "" (null parameter). An empty argument ("") is considered the same as if the argument was not specified.

```
comperr [-a][-w][-o][-e] [<drawing>] [<compile type>] [<context>]
```

This command runs ValidCOMPERR to print the compiler and linker error listings to a big output file. <drawing> is the name of the drawing to be viewed. It can be a root_drawing. If <drawing> is not specified, then the ROOT_DRAWING directive specified in the compiler.cmd file is used. <compile type> is the compilation type. If unspecified, ValidCOMPERR assumes LOGIC. The "-a" flag specifies that the user wants to always see all listings whether or not there are errors. The "-w" flag specifies that the user wants to see all severity messages of WARNING and higher (i.e. WARNINGS, or OVERSIGHTS or ERRORS). The "-o" flag specifies that the user wants to see all severity messages of OVERSIGHT and higher (i.e. OVERSIGHTS and ERRORS). The "-e" flag specifies that the user wants to see all ERROR messages only. If no flag is specified, the default flag is "-w". <context> specifies which context the drawing is to be looked at. Some example of <context> specifications: '1 version=4', 4, '2 delay=20', 'delay=40 version=2'. If <context> is to be specified, then the <drawing> and <compile type> arguments MUST be present as well (could be represented by an empty argument). Some example COMPERR commands:

```
comperr
```

Get any oversights, warnings, or errors found in any drawing under the ROOT_DRAWING specified in the compiler.cmd file since its last compile for the type also specified in the compiler.cmd file. The default context name is "0".

```
comperr -w counter
```

Get any oversights, warnings, and/or errors found in any drawing in the design "counter" since its last compile for the type specified in the compiler.cmd file. The default context name is "0".

```
comperr -o 'shifter' time
```

Get any oversights, and/or errors found in any drawing under the root drawing "shifter" since its last compile for TIME. The default context name is "0".

```
comperr -e '' '' '1 version=2 delay=40'
```

Get any errors found in the design and compile type which are specified in the compiler.cmd file and for context: SIZE=1, VERSION=2, and DELAY=40.

```
comperr -a "my design" sim 4
```

Get all listings (whether or not they have errors) found in all drawings of the design "my design" for the compile type SIM and for context SIZE=4.