V2V™ Frequently Asked Questions and Troubleshooting Guide

What if My Question isn't on this page?
License Related Questions
verilog2vhdl Related Questions
VHDL2Verilog Related Questions
General Questions

What if I have a question that isn't answered on this page?

If you run into a problem that does not have a solution on this page please send the following information, if available, to sales@syncad.com:

- A detailed description of the problem, including under what circumstances the problem occured, what version of the product you are using and on what platform you are using it.
- The log file: vhdl2v.log or v2v.log.
- The error message: cut and save the screen mesage to a file and attach it to your e-mail.
- The command line
- The input and output files

We will begin investigating the problem immediately, and respond to you as quickly as possible.

License Related Questions

I have installed ASC's HDL Translator on my Windows NT. When I run it, I keep receiving a license error "(ASC-LS) ERROR: Could not write in tmp-directory. Please check permissions." What shall I do?

The software uses two environment variables to verify the node-locked license : ASC_LICENSE_FILE and TMPDIR. This problem occurs if the TMPDIR is not set or is set to a non-writable directory. To solve the problem, either simply create a c:\tmp directory and make sure it is writable, or set the TMPDIR variable correctly by going to Start->Settings->Control Panel->System.

On Unix platform, when I run "lmstat -a" to check the status of the license server, I receive an "unsupported by licensed server" error. What can I do?

ASC's license deamon (ASCLMD) requires FlexLM 6.0 or newer. If you have old FlexLm installed on your system, you may have invoked an old version of lmgrd. You can check the version by runnning

```
lmgrd -v
```

If your version is older than 6.0 version, then you should use ASC supplied lmgrd from the `$ASC_HOME/license` directory.

What environment variables are needed for Unix users?

For Unix platform, two variables are requied to set before running the Translators: ASC_HOME and ASC_LICENSE_FILE. ASC_HOME is the directory where you installed the ASC's tools, such as: /usr/local/ASC. ASC_LICENSE_FILE is the license file you received from ASC, such as /usr/local/ASC/license/asc.lic. These two environment settings need to be added to user's startup file by the user.

I am using Windows NT. How do I find the hostID?

For Windows NT system, ASC's tools take Disk_Serial_Number as hostID. If you have installed the software, you can run **Lmtool** from the Start menu. When the interface appears, click on the **HostID** button. You will see the **DISK_SERIAL_NUM**. Send that information to ASC's to obtain your license. You can also get the Disk_Serial_Number without Lmtool, by doing a **DIR** command on a DOS prompt. You will see the **Volume Serial Number** on the second line of the content.

---

verilog2vhdl Related Questions

Can v2v™ handle compiler directive 'include?

Yes. The 'include directive is handled by Vpp, the "Verilog Pre-Processor." It is in the "bin" subdirectory after you install the Translator.

Vpp deals with some of the compiler directives such as 'include. The output file of Vpp is still a Verilog file, which may be translated automatically by the verilog2vhdl translator.

In most cases Vpp is unnecessary. It is only needed when you have compiler directives in your original Verilog file.

I received a warning message during translation that I used a reserved word in my code, or the identifier is not a legal VHDL identifier. What does it mean?

ASC's translator has a list of reserved words, such as "input" "output" "mod", etc, that can not be used as identifiers. Also, some legal identifiers in Verilog are not legal in VHDL, such as an identifier like "name_".

The Translator treats this case as a warning. It adds a slash "/" to both ends of the identifier, such as "/name_/", and gives out a warning message to remind the user.

Please look at verilog2vhdl and vhdl2verilog manuals as a part of this documentation to see a complete list of reserved words.

Can v2v™ translate modules with multiple levels of instantiation?

Yes!

Suppose you have a module A (in file "file1.v") that instatiates module B (in file "package1.v") and C ("package2.v"). In this case, you would use the following command line:

```
v2v file1.v file1.vhd -p package1.v package2.v
```

Now consider a situation in which module B also instantiates module D (in file "package3.v"). In this case you would use the following command line:

```
v2v file1.v file1.vhd -p package3.v package1.v package2.v
```

Please note, in the second case, the sequence of the package files matters. You have to put the lowest level module first, then the higher level modules.

In some situations you might not want to check the existence of some lower level modules while you are translating the top level module. Use the `-nc` option to acheive this. You can see all the options with the command `v2v -h` or by reading the users' manual.

How does v2v™ perform error handling?

v2v™ handles errors in the following ways:

1. If a syntax error in encountered in the input file, the tool exits immediately after printing out the error message.
2. If an unsupported construct is encountered in the input, a warning message is issued, and v2v™ continues processing.
3. If the unsupported construct is of the type `event', v2v™ exits after issuing an appropriate error message.

What Verilog constructs are *not* supported?

- UDPs
- Assign and deassign procedural assignments
- Force and release procedural assignments
- Parallel blocks (such as fork-join blocks)
- Task disable
- Specify blocks
- There are some system functions not supported, such as the ones for timing check.

The supported and unsupported constructs are listed in detail in the user's manual, as well as some workarounds to common problems.

How do I verify the translation result?

You can use any simulation tool to verify the functionality of a translated file.

v2v™ keeps functional equivalence of the Verilog design and the translated VHDL design. It is intended to be used to enhance productivity by saving much of the labor of manual translation. Some language constructs may not be translated automatically due to language disparities.

All the library files needed for the simulation are included in the package. When you compile these library files, please follow the instructions about the sequence in the v2v™ Manual (v2v_man.html), in the section entitled "post-translation".

I use the Synopsys synthesis and simulation environment. Is there anything I need to take into consideration?

Package files compliant with the SYNOPSYS simulation and synthesis environment are located in the *$ASC_HOME/libsyn* directory. Filenames are almost the same as described below; the only difference being that files in the *$ASC_HOME/libsyn* have a '_syn' extension. Therefore, the file buffers.vhd in the *$ASC_HOME/lib* is equivalent to file *buffers_syn.vhd* in the libsyn directory.

Why do I receive compilation errors when I compile translated VHDL files?

In some cases, the automatic translation can not complete due to unsupported constructs. Hand modification is required to make the translation compilable.

Sometimes you receive compilation errors because some simulation/synthesis tools have problems with supporting IEEE1076-1993. Use ***_87 instead. The readmem.vhd, textio_header.vhd, and standard.vhd are compliant with VHDL-93. If you must use them, please refer to the simulation/synthesis tool's manual and modify the files to make them compatible to the tool.

VHDL2Verilog Related Questions

What does the `-ssc` switch do?

`-ssc` means **S**upport **S**ubprogram **C**all. It causes the translator to translate subprogram calls in the header only. You will see a verilog header in the <package_name>_package.verilog file, but no subprogram body. This is for the situations where some special functions are desired to be tranlsated manually.

I can not find the std_logic_misc or std_logic_signed packages in your library. DO you provide it?

ASC does not provide these packages. You can find them in many simulators, but some of the packages use v2v 's reserved words. The best way to work this out is to tailor the standard packages to your need. In most cases you only need a small part of the functions in the packages. You copy the standard package file to your own directory, delete the functions you do not need, and save it as your own file. Put this file name in the vhdl2v.map file instead of the standard package then do the translation. Even if the translator complains about the reserved words in the package, you can easily fix it because it is very small.

What does the `-ssb` switch do?

SSB stands for **S**upport **S**ubprogram **B**ody.

`-ssb` is an extension of `-ssc`. The translator translates the subprogram bodies as well as the headers. In this case, the whole package will be read in by the Translator even though only one or two functions in the package are used.

Does the tool append new information to <package_name>_package.verilog every time a new design file is converted, or is the existing package data in <package_name>_package.verilog overwritten with new data from the most recently converted file?

The new <package_name>_package.verilog will overwrite the old one, because each <package>.verilog is exclusively for the <output>.verilog file. The package is embedded to the <output>.verilog by an 'include statement.

Does the tool support the user naming the output file for converted packages to something other than the default <package_name>_package.verilog?

The tool cannot change the package name, but you can. Just rename the package file (or even edit it as you like), and change the corresponding name in the 'include statement in the <output>.verilog.

What should I do if I receive a `Construct not supported` warning message?

This warning message means that in your code you have used constructs that ASC's Translator does not support at present time. This is because VHDL and Verilog cover different levels of abstraction, and some constructs in one do not have matching constructs in the other. In other words, "translatable" VHDL/Verilog is only a subset of standard VHDL/Verilog. Not all the constructs can be automatically translated. The tool will leave a comment in the place of the unsupported constructs in the output file to remind user to hand translate them.

The major unsupported constructs are

- generics without a default value
- signal assignments with multiple waveforms
- unconstraint types and their attributes
- multidimensional accesses
- access type
- concurrent procedure calls
- TEXTIO procedures and functions

For a detailed list of unsupported constructs in VHDL and Verilog, please refer to the manual bundled with the software package.

What do the WARNINGs mean? Can I ignore them?

The warning messages mean that the resultant translation may or may not be correct. The warning messages remind you to check those lines of the translation

to make sure they are correct. If you are sure, just ignore the warnings. If not, run the translation through a simulator to see if it complains about the indicated lines.

I received an error message during translation that I used a reserved word in my code. What does it mean?

ASC's translator has a list of reserved words, such as "input" "output" "mod", etc, that can not be used as identfiers.

The solution is to globally search and replace the reserved words you used with other names. A list of reserved words can be found in the Table of Reserved Words and in the release notes.

What is function mapping?

The primary purpose of the function mapping file is to direct VHDL2verilog on how it should process FUNCTIONs present in input VHDL.

It can be used to suppress printing of type conversion function calls widely used in VHDL. Most of the type conversions are not needed in Verilog, and VHDL2verilog can read in a user-supplied function mapping file(s), and utilize information therein to create optimized Verilog output.

In addition, a function map file can be used to specify a replacement string that will replace all occurences of a function in the VHDL input. Each function can have its own string. It is assumed that the object specified by the replacement string exists.

How do I use function mapping files to direct the translation of a function?

When you translate a VHDL file, you can ignore VHDL functions or replace VHDL functions with the ones you specified. A function mapping file will tell the translator what functions should be ignored, or what functions should be replace. You can modify the function mapping file according to your requirements. When you perform the translation, add an option `-fm mapping_file_name` to the end of the command line.

For detailed information of using function mapping file, please refer to User's Manual.

General Questions

How do I install a patch?

Every so often patches to the software will be made available. These will be distributed from ASC's ftp-site (ftp.ascinc.com). Once an appropriate patch has been retrieved from the ftp-site, move it one level up from the $ASC_HOME directory, and start the install script followed by a period ('.'):

1. Get a patch from ASC's ftp-site
2. **cd $ASC_HOME; cd ..**
3. **../install_pgm .**
4. Follow the instructions in the installation script

What if some of my comments are missing or misplaced in the translated version of my code?

We realize that our translators do not always preserve comments perfectly. This is because the translation is not line-by-line, so it can be difficult to place the comment strings in the resultant code. For example, the concurrent assignments in VHDL may be translated to Verilog in different order, so the translator may not be smart enough to decide if the comment should go with the first statement or the second one. Besides different syntax of VHDL and Verilog statements make it difficult for the tool to associate comments properly with the line before or after the comment.

The aim of the translator is to translate code from one HDL to the functional equivalent of the other (same simulation and synthesis results). The current version has an option to suppress all comments, which will solve your problem if the misplaced comments are merely an annoyance.

In the latest release several enhancements have been made to keep comments in place. We will keep working on the issue and users will see the further improvement gradually.

What if My Question isn't on this page?
License Related Questions
verilog2vhdl Related Questions
VHDL2Verilog Related Questions
General Questions
