You are here: Synthesis Man Pages > **Synthesis Tool Commands** > **a** > analyze

analyze

NAME

analyze

Analyzes the specified HDL source files and stores the design templates they define into the specified library in a format ready to specialize and elaborate to form linkable cells of a full design.

SYNTAX

```
status analyze
    [-format verilog | sverilog | vhdl]
    [-define define_list]
    [-library library_name | -work library_name]
    [-uses design_libs_list]
    [-vcs vcs_opts]
    [-create_update]
    [-update]
    [-recursive]
    [-autoread]
    [-rebuild]
    [-output_script output_string]
    [-exclude exclude_list]
    [-verbose]
    [-top top_design_name]
    file_list
```

Data Types

define_list	string
library_name	string
design_libs_list	string
vcs_opts	string
output_string	string
exclude_list	list
top_design_name	string
file_list	list

ARGUMENTS

-format verilog | sverilog | vhdl

Specifies the format of the files to be analyzed. The supported formats are Verilog, SystemVerilog, and VHDL.

Specifies that the **-autoread** option should look for files in the specified language whose extensions match the extensions in the list identified by the **hdlin_autoread_verilog_extensions**, **hdlin_autoread_sverilog_extensions**, or **hdlin_autoread_vhdl_extensions** variable. The **-autoread** option supports Verilog, SystemVerilog, and VHDL formats. By default, **-autoread** reads in all the files it finds that have Verilog, SystemVerilog, and VHDL extensions. You can use the **-autoread** option to specify file names in the *file_list* that do not have one of the language-specific extensions.

-define *define_list*

Specifies a list of constant macros to be defined for the session of the **analyze** command. This is useful for the inclusion of Verilog or SystemVerilog code enclosed by `` `ifdef/`endif `` constructs. For details, see the *HDL Compiler for Verilog User Guide*.

-library *library_name*

Specifies *library_name* as the current work library (whose **-path** receives all analyzed output files). The *library_name* should be previously established by the **define_design_lib** command. Design libraries (directories containing template intermediate files) should not be confused with .db or .ddc linkage libraries (files containing design cells).

By default, the **analyze** command stores all output in the work design library. To store design elements in libraries other than work library, use the **-library** option. A writable directory must exist at the path defined for the current work library.

-work *library_name*

An alias for **-library**. This option is not available with the **-autoread** option.

-uses *design_libs_list*

This option is not available with the **-format vhdl** or **-autoread** option.

SystemVerilog and Verilog designs might contain unresolved references to unelaborated design templates without indication of where those template definitions are located.

The compiler and linker always search for these templates first in the home directory of the parent design. By default, they then continue each template search by visiting every design library defined by the **define_design_lib** command. The **-uses** option overrides this last step. It stores a design library search order (as a cell attribute named **link_design_libraries**) within each design being analyzed. Later, elaboration or linking will follow this prescribed order rather than visiting every known library. An empty list argument limits the search to just the home library of the parent design. You do not need to repeat the **-lib** setting, and you cannot override a search of the work directory.

-vcs *vcs_opts*

Specifies all VCS-specific command-line options. The options must be enclosed with double quotation marks (""). Options specified by the **-vcs** option follow the VCS command-line syntax:

[-sverilog | -verilog] Specifies the format of the files to be analyzed.

[-y *directory_path*] Specifies the directories that contain the library files to be searched for unresolved module instantiations in the design.

[+libext+*.extension1*+...] Specifies the extension to consider during a file search in the -y library directories. The default is no extension.

[-v *library_file*] Holds several module definitions to be used during the search for unresolved modules.

[-f *command_file*] Specifies a command file.

[+define+*macro_name*+...] Defines a macro.

[+incdir+*dir1*+...] Includes directories in the search list.

src_file1 Specifies the files that are to be analyzed.

This option is not available with the **-autoread** option.

-create_update

Facilitates the packaging of HDL files for distribution performed by developers of DesignWare parts. For details, see the *DesignWare Developer's Guide*. This option is not available with the **-autoread** option.

-update

Facilitates the installation of DesignWare parts in DesignWare installation scripts. For details, see the *DesignWare Developer's Guide*. This option is not available with the **-autoread** option.

-recursive

Specifies that the **analyze -autoread** command must recursively scan the directory trees below those specified in the *file_list* list. By default, the **-autoread** option looks only in the specified directories.

This option can be used only with the **-autoread** option.

-autoread

Specifies to use the **-autoread** mode for script-free analysis of whole HDL source directories or directory trees. All HDL source files located in those directories are prescanned, grouped, and then ordered into distinct source code streams based on their mutual inclusion and package-reference dependences. Some or all source streams are finally analyzed using the basic command.

By default, the **-autoread** option creates or replaces only a subset of analyzed result files; it works in an incremental-update mode that resembles the Unix make command. If an HDL source file does not exist or is newer than the intermediate files it should produce, only that file and the source streams that require it are reanalyzed.

For more information, see the description for the **-rebuild** option.

-rebuild

Replaces any result files regardless of their timestamp. It analyzes all the indicated HDL source streams as if the result library was empty.

This option can be used only with the **-autoread** option.

-output_script *output_string*

Creates a Tcl script that is compatible with Design Compiler and Formality.

The file reading order for third-party tools and a DOT language file dependency graph are embedded in the script as comments.

This option is available only with the **-autoread** option.

-exclude *exclude_list*

Specifies a list of files and directories that are not to be analyzed. An **analyze -autoread [...] file_list** command checks each HDL source path against all elements of the *exclude_list*; it ignores a file if it or any of its directory paths matches any excluded path.

This option can be used only with the **-autoread** option.

-verbose

Prints out more messages for the **-autoread** option.

This option can be used only with the **-autoread** option.

-top *top_design_name*

Identifies the top-level component of a hierarchical design specified within the **-autoread** HDL source directories.

The **analyze -autoread -top** *design name* command selects a subset of the indicated HDL source streams to analyze. Its goal is to analyze into intermediate form precisely those source files required for a later elaboration and linking of *design_name*. Files in the indicated directories that are not required to elaborate the design hierarchy descending from *design_name* are prescanned but the design templates they would produce are not created or replaced.

If the **-top** option is not provided, all HDL source streams indicated by the **file_list** argument are analyzed.

This option can be used only with the **-autoread** option.

file_list

Specifies the files to be analyzed.

To analyze multiple files, list them by using curly braces, {}. Note: Source files in a list are effectively concatenated into a single source stream in the listed order.

In **-autoread** mode only, the *file_list* argument can also specify directory names. All HDL source files located in those directories are prescanned. For more information about prescanning, see the descriptions for the **-autoread** and **-recursive** options.

DESCRIPTION

This command translates the specified HDL files and stores the intermediate format in the specified library.

The following paragraphs describe the support for the **-autoread** option:

The **-autoread** option reads in the files from file_list, determines the dependencies between them, and analyzes them in the right order to prevent errors due to missed or misplaced files.

The dependencies are calculated only from the files or directories present in *file_list*. If *file_list* changes between consecutive calls with the **-autoread** option, the dependency inference is performed over the latest set of files provided. Based on this, all required sources must be present in *file_list* option or be available if the **-recursive** option is used on each call with the **-autoread** option.

If the top design name is provided by the **-top** option, only the source files required for elaborating that top design are analyzed. This filtering is based on file dependencies previously inferred. If the **-top** option is not provided, all sources are ordered, grouped, and analyzed per the dependency inferred between them.

To locate the source files, the command expands each item in the *file_list* list with the **search_path** variable. Then, it determines whether the result is excluded by the **-exclude** option or the **hdlin_autoread_exclude_extensions** variable. If it is not excluded and a file ends with one of the extensions in the **hdlin_autoread_vhdl_extensions** variable extensions list, it is considered a VHDL source file. If the file ends with one of the extensions in the **hdlin_autoread_verilog_extensions** variable, it is considered a Verilog source file. If the file ends with one of the extensions in the **hdlin_autoread_sverilog_extensions** variable, it is considered a SystemVerilog source file.

When expanding a directory, the command collects the files from the directory, and from its subdirectories and theirs recursively if the **-recursive** option is set. The command then performs all extension checks on these files. If the **-format** option is set, only files with Verilog, SystemVerilog, or VHDL extensions are collected based on the value of the option.

After the **-autoread** option collects all of the source files, it performs the following dependency checks:

- Detects analyze dependencies: List RTL files in the right order for analyzing. For example, analyze the file that contains a VHDL entity before analyzing files that defines architectures of that entity, or analyze the file that contains a SystemVerilog package declaration before the files that import that package.
- Detects Verilog and SystemVerilog compilation unit dependencies: Determine if a file needs to be analyzed in the same compilation unit with other files. For example, if a file defines macros, SystemVerilog local parameter, and SystemVerilog enumerated values, and the file is not explicitly included by the file that requires that definitions, the **-autoread** option groups them in the same compilation unit in the correct order. This might not always be possible, such as when a macro is defined several times in different files and the **-autoread** option cannot determine which of those alternatives is the right choice.
- Detects link dependencies: Schedule the analyze stage for files required for elaboration of the design hierarchy. For example, if a Verilog or SystemVerilog design that is instantiated in one source file is

declared on a different source file provided in the *file_list*, the second file also requires to be analyzed for a complete top-down design elaboration.

- Detects include dependencies: If a Verilog or SystemVerilog file is included in another source and the file changes between two consecutive calls of the **-autoread** option (in the same DCSHELL session), the **analyze** command includes this file and all files that include them to update the design.
- Infers the target library for VHDL files. However, if the **-library** option is specified, this step is skipped and VHDL files provided in the *file_list* are analyzed into the specified design library.

After the dependency check, all the required HDL files specified by the *file_list* option (regarding its dependencies and the **-top design** option if defined) are analyzed.

When the **-autoread** option is used again (with the same *file_list* setting) after a file is changed, only the updated source files are analyzed.

The **-autoread** option executes the dependency analysis based only on the current *file_list* information. All HDL source files that might have relevant dependency relationships should be passed together in one **analyze -autoread** command.

EXAMPLES

The following example analyzes the packages.vhd file and stores the results in the MY_LIB design library:

```
prompt> analyze -work MY_LIB -format vhdl packages.vhd
```

The following example analyzes two SystemVerilog clients of two libraries:

```
prompt> define_design_lib BUSDEFN -path /remote/IP/metra/big_bus
prompt> define_design_lib NClib   -path ../../NCD/libNC
prompt> analyze -format sverilog -uses { BUSDEFN  NClib }  { senderAlice.sv receiverBob.sv }
```

The following example assumes that the current directory is the source directory. It specifies the source file list early in the command line and gives options, including the name of the top-level entity later.

```
prompt> analyze {.} -autoread -recursive -top E1
```

The next example specifies extensions for Verilog files that are different from the default ({.v}), sets the source list and the exclude list, and calls the command with the name of the top-level module name, forcing it to only collect files with Verilog extensions:

```
prompt> set hdlin_autoread_verilog_extensions {.ve .VE}
prompt> set my_sources {mod1/src mod2/src}
prompt> set my_excludes {mod1/src/incl mod2/src/incl}
prompt> analyze $my_sources -exclude $my_excludes -autoread \
   -format verilog
```

Note that excluding include directories explicitly is only necessary if include files have the same extensions as source files and not all include files are included in the source.

SEE ALSO

define_design_lib(2)
elaborate(2)
link(2)
read_file(2)
report_design_lib(2)
hdlin_autoread_exclude_extensions(3)
hdlin_autoread_verilog_extensions(3)
hdlin_autoread_sverilog_extensions(3)
hdlin_autoread_vhdl_extensions(3)