

Design Compiler® Automated Test Case Packaging Application Note

Version G-2012.06, June 2012

SYNOPSYS®

Copyright Notice and Proprietary Information

Copyright © 2012 Synopsys, Inc. All rights reserved. This software and documentation contain confidential and proprietary information that is the property of Synopsys, Inc. The software and documentation are furnished under a license agreement and may be used or copied only in accordance with the terms of the license agreement. No part of the software and documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Synopsys, Inc., or as expressly provided by the license agreement.

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Synopsys, AEON, AMPS, ARC, Astro, Behavior Extracting Synthesis Technology, Cadabra, CATS, Certify, CHIPit, CODE V, CoMET, Confirma, Design Compiler, DesignSphere, DesignWare, Eclipse, Formality, Galaxy Custom Designer, Global Synthesis, HAPS, HapsTrak, HDL Analyst, HSiM, HSPICE, Identify, Leda, LightTools, MAST, MaVeric, METeor, ModelTools, NanoSim, NOVeA, OpenVera, ORA, PathMill, Physical Compiler, PrimeTime, SCOPE, SiVL, SNUG, SolvNet, Sonic Focus, STAR Memory System, SVP Café, Syndicated, Synplicity, Synplify, Synplify Pro, Synthesis Constraints Optimization Environment, TetraMAX, the Synplicity logo, UMRBus, VCS, Vera, and YieldExplorer are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, ASAP, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, BEST, Columbia, Columbia-CE, Cosmos, CosmosLE, CosmosScope, CRITIC, Custom WaveView, CustomExplorer, CustomSim, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, DesignPower, DFTMAX, Direct Silicon Access, Discovery, Encore, EPIC, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, High-performance ASIC Prototyping System, HSiM^{plus}, i-Virtual Stepper, IC Compiler, IICE, in-Sync, iN-Tandem, Intelli, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Macro-PLUS, Magellan, Mars, Mars-Rail, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, MultiPoint, ORAengineering, Physical Analyst, Planet, Planet-PL, Platform Architect, Polaris, Power Compiler, Processor Designer, Raphael, RippledMixer, Saturn, Scirocco, Scirocco-i, SiWare, SPW, Star-RCXT, Star-SimXT, StarRC, Symphony Model System Compiler, System Compiler, System Designer, System Studio, Taurus, TotalRecall, TSUPREM-4, VCSi, VHDL Compiler, VMC, and Worksheet Buffer are trademarks of Synopsys, Inc.

Service Marks (SM)

MAP-in and TAP-in are service marks of Synopsys, Inc.

Third Party Trademark Acknowledgements

SystemC is a trademark of the Open SystemC Initiative and is used under license.

ARM and AMBA are registered trademarks of ARM Limited.

Saber is a registered trademark of SabreMark Limited Partnership and is used under license.

Entrust is a registered trademark of Entrust Inc. in the United States and in certain other countries. In Canada, Entrust is a trademark or registered trademark of Entrust Technologies Limited. Used by Entrust.net Inc. under license.

All other product or company names may be trademarks of their respective owners.

Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Contents

About This Application Note	vi
Customer Support.	viii
1. Using the Automated Test Case Packaging Utility	
Setting the SNPS_ENABLE_TESTCASE Variable and Running the Test Case . . .	1-2
Packing the Test Case	1-3
Unpacking the Test Case	1-4
File Structure of the Extracted Data	1-6
Reproducing the Test Case.	1-7
Limitations.	1-8

Preface

This preface includes the following sections:

- [About This Application Note](#)
- [Customer Support](#)

About This Application Note

Design Compiler provides an automated test case packaging utility for debugging and benchmarking purposes. The `dc_pack_design` utility allows you to run a script that packs your designs and files, regenerates the command script used in your Design Compiler session, and creates a tar file that runs in the Synopsys environment to reproduce the test case. This utility alleviates the time-consuming task of packing your design and makes it easier to reproduce the test case in the Synopsys environment. To unpack and reproduce the test case, use the `dc_unpack_design` utility.

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
Courier	Indicates syntax, such as <code>write_file</code> .
<i>Courier italic</i>	Indicates a user-defined value in syntax, such as <code>write_file design_list</code> .
Courier bold	Indicates user input—text you type verbatim—in examples, such as <code>prompt> write_file top</code>
[]	Denotes optional arguments in syntax, such as <code>write_file [-format fmt]</code>
...	Indicates that arguments can be repeated as many times as needed, such as <code>pin1 pin2 ... pinN</code>
	Indicates a choice among alternatives, such as <code>low medium high</code>
Ctrl+C	Indicates a keyboard combination, such as holding down the Control key and pressing C.
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

SolvNet includes a knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. SolvNet also gives you access to a wide range of Synopsys online services including software downloads, documentation, and technical support.

To access SolvNet, go to the following address:

<https://solvnet.synopsys.com>

If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.

If you need help using SolvNet, click HELP in the top-right menu bar.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a support case to your local support center online by signing in to SolvNet at <https://solvnet.synopsys.com>, clicking Support, and then clicking “Open A Support Case.”
- Send an e-mail message to your local support center.
 - E-mail support_center@synopsys.com from within North America.
 - Find other local support center e-mail addresses at <http://www.synopsys.com/Support/GlobalSupportCenters/Pages>
- Telephone your local support center.
 - Call (800) 245-8005 from within North America.
 - Find other local support center telephone numbers at <http://www.synopsys.com/Support/GlobalSupportCenters/Pages>

1

Using the Automated Test Case Packaging Utility

Design Compiler provides an automated test case packaging utility for debugging and benchmarking purposes. The `dc_pack_design` utility allows you to run a script that packs your designs and files, regenerates the command script used in your Design Compiler session, and creates a tar file that runs in the Synopsys environment to reproduce the test case. This utility alleviates the time-consuming task of packing your design and makes it easier to reproduce the test case in the Synopsys environment. To unpack and reproduce the test case, use the `dc_unpack_design` utility.

This chapter includes the following sections:

- [Setting the SNPS_ENABLE_TESTCASE Variable and Running the Test Case](#)
- [Packing the Test Case](#)
- [Unpacking the Test Case](#)
- [File Structure of the Extracted Data](#)
- [Reproducing the Test Case](#)
- [Limitations](#)

Setting the SNPS_ENABLE_TESTCASE Variable and Running the Test Case

Before you package a test case in the Design Compiler session, you need to set the `SNPS_ENABLE_TESTCASE` UNIX environment variable. After setting this variable, you run the `dc_shell` command to run the test case. This document uses the `run_dc.tcl` file as the test case. It is the top-level script from the initial run.

Follow these steps to set the `SNPS_ENABLE_TESTCASE` variable and run the test case:

1. Set the `SNPS_ENABLE_TESTCASE` UNIX environment variable:

```
% setenv SNPS_ENABLE_TESTCASE
```

2. Run the `dc_shell` command and specify the test case to be packaged:

```
% dc_shell -topographical -f run_dc.tcl
```

After running the `dc_shell` command, two log files are generated during the Design Compiler session: The `command_jobid_time_stamp.log` file includes the `dc_shell` setup file commands and variable assignments. The `filenames_jobid_time_stamp.log` file consists of all the files that were used in the Design Compiler session. When Design Compiler generates the log files, it appends the process ID and timestamp to the files. For example,

```
Initializing...
Starting shell in Topographical mode...
Command log file for this session
'command_13245_D20110713.log'
Filenames log file for this session
'filenames_13245_D20110713.log'
```

To run the `dc_pack_design` utility to create the test case package, see [“Packing the Test Case.”](#)

Packing the Test Case

The `dc_pack_design` utility, which is located in the Synopsys root directory, packs your designs and the files used in your Design Compiler session and creates a tar file. The test case package consists of all the files that were captured in the command log and filenames log files when you ran the test case. You need to run this utility in the same directory as your Design Compiler session.

To package a test case, specify the command log and filenames log files by using the `-c` option and `-f` option respectively with the `dc_pack_design` utility. For example,

```
% dc_pack_design -c command_13245_D20110713.log \
                  -f filenames_13245_D20110713.log
```

Then choose one of the following options:

- To specify the name of the archive file for the test case, use the `-o` option. For example,

```
% dc_pack_design -c command_13245_D20110713.log \
                  -f filenames_13245_D20110713.log \
                  -o my_testcase.tar.gz
```

If you do not specify the `-o` option, the default name of the archive file is `SNPS_TESTCASE.tar.gz`.

- To add multiple files to the package in addition to the files listed in the filenames log file, use the `-l` option.

```
% dc_pack_design -c command_13245_D20110713.log \
                  -f filenames_13245_D20110713.log \
                  -l "/u/proj1/work/output.log /u/proj1/lib/new_lib.db"
```

- To remove files from the archive file or add additional files, use the `-t` option and follow these steps:

- Run the `dc_pack_design` utility with the `-t` option:

```
% dc_pack_design -c command_log_filename.log -f \
filenames_log_filename.log -t run_tar.sh
```

This creates a `run_tar.sh` file, which you will later source to create the archive file.

In the current directory, you will find the `snps_archive_list` file. This list contains all the files that will be archived by the tar command file.

- Modify the `snps_archive_list` file by removing the files you do not want to include in the archive file or by adding any new files that you want to include in the archive file.
- Source the tar command file to create the new archive file:

```
% sh -x run_tar.sh
```

- d. After you create the archive file, remove the SNPS_PACK_DESIGN* temporary directory and the snps_archive_list file.

For more information about the available options for the `dc_pack_design` utility, use the `-help` option. For example,

```
$SYNOPSYS/$TARGET_ARCH/syn/bin/dc_pack_design -help
```

Unpacking the Test Case

To unpack the test case, use the `-f` option with the `dc_unpack_design` utility to specify the archived file. You can run the utility in any directory. However, if the directory does not contain the SNPS_TESTCASE.tar.gz tar file or the test case tar file that you renamed, you must specify the path. For example,

```
% dc_unpack_design -f to_path/my_testcase.tar.gz
```

By default, the directory where you run the `dc_unpack_design` utility becomes the new root directory for the test case. The utility unpacks the archive file and creates the `snps_readme.txt` file, which contains the instructions for reproducing the test case and the list of files that were unarchived. The utility also creates the `snps_run.csh` template C-shell script that is used to reproduce the test case.

When you unpack the archive file, data is extracted into the directory structure described in [“File Structure of the Extracted Data” on page 1-6](#).

The following options are available to run the `dc_unpack_design` utility:

- (Optional) To unpack the test case in a specific directory instead of the current working directory, use the `-to` option. For example,

```
% dc_unpack_design -to testcase1_directory
```

- To specify a test case other than the default SNPS_TESTCASE.tar.gz tar file to unpack, use the `-f` option. If you use the `-o` option to specify an archive file, use the same specified file for the `-f` option. For example,

```
% dc_unpack_design -f my_testcase.tar.gz
```

- (Optional) To specify a file name containing the extensions that should be skipped for path modifications, use the `-ignore_extensions` option.

By default, files with the following extensions are not modified when you run the `dc_unpack_design` utility: `.def`, `.scandef`, `.v`, `.vhdl`, `.hdl`, `.sv`, `.saif`, `.ctl`, `.sdf`, `.spef`, `.tf`, `.map`, `.tluplus`, `.tlu`, `.tlu_plus`, and `.tlup`.

If you want to change this default, you can create a new file that contains the extensions to be skipped for path modifications and specify it by using the `-ignore_extensions` option. For example, you can create an `ignore.txt` file that includes the `.v`, `.tf`, and `.gv` extensions. Run the following command so that the `dc_unpack_design` utility does not modify files with these extensions:

```
% dc_unpack_design -ignore_extensions ignore.txt
```

The `dc_unpack_design` utility performs the following tasks:

- Replaces the absolute path names in the unarchived files with the current directory name where the test case package was extracted

In the top-level script from the original environment and the `snps_run.tcl` script, the following Tcl variable specifies the new location of the test case:

```
set SNPS_TESTCASE_ROOT /u/testcase/project/reproduce
```

The absolute path names in the original files are replaced using this variable.

- Adds environment variables that are used to run the test case in the main file of the original scripts and the `snps_run.tcl` script

Note:

The original file is stored as *filename.snps.bak* when the file is modified by the `dc_unpack_design` utility. You can find the list of the files that are modified by the `dc_unpack_design` utility in the `snps_readme.txt` file.

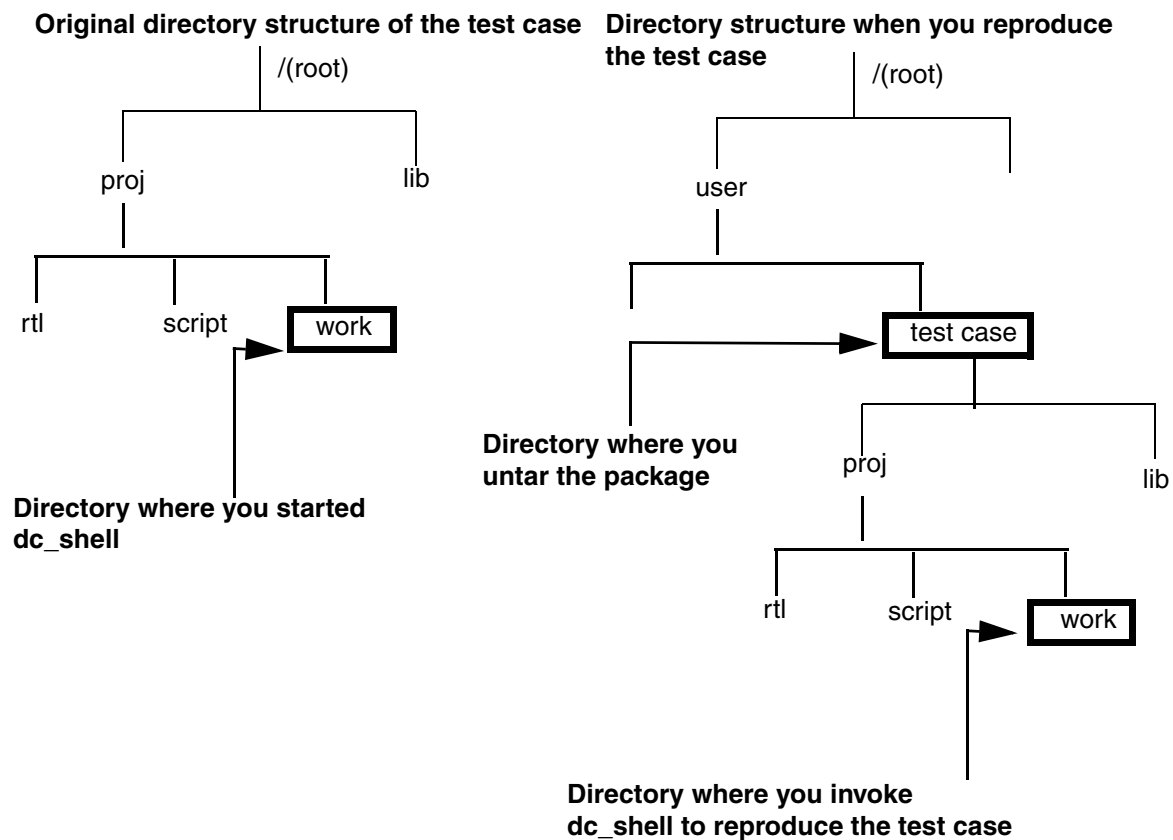
For more information about the available options for the `dc_unpack_design` utility, use the `-help` option. For example,

```
$SYNOPSIS/$TARGET_ARCH/syn/bin/dc_unpack_design -help
```

File Structure of the Extracted Data

When you untar the archive file, the directory structure of the extracted data is as shown in [Figure 1-1](#).

Figure 1-1 File Structure of the Extracted Data



Important:

When you extract the archive file, the original directory structure of the test case is replicated in your environment. To reproduce the test case, you must invoke `dc_shell` in the same directory where you ran `dc_shell` in the original environment, as shown in [Figure 1-1](#).

In addition to the directories shown in [Figure 1-1](#), the following files and directories are generated in the directory where you untar the package:

- `snps_readme.txt`

This file contains information about how to reproduce the test case—that is, information such as the top-level run script, the `dc_shell` run mode, and the directory to source the run script.

- **SNPS_FILE**

This directory contains the setup files, command log, and filenames log file.

The following files store the setup files sourced in the `dc_shell` session in the original environment:

- **ROOT.synopsys_dc.setup**

This is the setup file from `$SYNOPSYS` root.

- **LOCAL.synopsys_dc.setup**

This is the setup file from the local directory.

- **USER.synopsys_dc.setup**

This is the setup file from the home directory.

- **USER_FILE**

This directory contains the files that are added by using the `-l` option.

The following files are created in the directory where you run the test case:

- `snps_run.tcl`, the tool-generated Tcl replay script
- `snps_run.csh`, the shell script used to reproduce the test case
- `snps_user_env_vars.tcl`, the set of UNIX environment variables

Reproducing the Test Case

You can use either of the following two methods to reproduce the test case. In both methods, the path names of all the files are automatically modified by the `dc_unpack_design` utility:

- Using the top-level script from the original environment, for example, `run_dc.tcl`

The `dc_pack_design` utility copies all the files sourced from the top-level script from the original run, allowing you to reproduce the test case the same way.

- Using the `snps_run.tcl` script, which is automatically generated by the tool
 - This is the `snps_run.tcl` script that is created by the `dc_pack_design` utility. All the commands in the setup files, all the commands in the top-level script, and all the commands in all the files sourced from the top-level script are included in this script.

If a Tcl package is used in the original environment, all the contents are duplicated in the `snps_run.tcl` script. You must remove the included package contents from the `snps_run.tcl` script.

For example, remove the following contents:

```
Package if needed snpsTclPro 1.0 [list source [file
  join $dir snpsTclPro.tcl]]
# -- End source /package_dir/tcllib/snps_tcl/snpsTclPro/
pkgIndex.tcl
Design Compiler 2011.09 Release - Automated Design Packaging
```

- The script appends all sourced scripts into a single file.
- When the script has a complex structure, such as nested if statements and many procedures, manual modifications might be needed.

To modify the script and reproduce the test case,

1. Go to the directory where you ran `dc_shell`. See the `snps_readme.txt` file in the directory where you unpacked the test case to find out which directory to go to, for example:

```
% cd project/work
```

2. Open the `snps_run.csh` script and modify the script if needed to choose between using the original script (for example, `rund_dc.tcl`) or using the tool-generated script (`snps_run.tcl`) to reproduce the test case.

The `snps_run.csh` shell script contains the following two `dc_shell` command examples:

- If you are using the original script, run the following command:

```
% dc_shell -no_home_init -no_local_init -topographical -f run_dc.tcl
```
- If you are using the `snps_run.tcl` script, run the following command:

```
% dc_shell -no_home_init -no_local_init -topographical \
-f snps_run.tcl
```

Note:

You must use the `-no_home_init` and `-no_local_init` options for either command.

3. Run the `snps_run.csh` script to reproduce the test case. For example,

```
% snps_run.csh
```

Limitations

The `dc_pack_design` utility has the following limitations:

- The archive file does not include
 - Files sourced by the `sh` command.

- VHDL packages that are not analyzed in the `dc_shell` session in the original environment.
- Files opened with the `open` command.
- Files checked by the `file` command.
- If there are any commands that write to files that are not in the current directory where the top-level script is run, you need to replicate the appropriate directories; otherwise, the synthesis run will have errors.

In the following example, the `./db` directory is created before you invoke `dc_shell` in the Synopsys environment:

```
dc_shell> write -format db -o ./db/TOP.db
```

Other directories that might not be in the current directory and should also be replicated include report directories, output-constraint directories, and output-netlist directories.

- When the RTL file contains an include file with the full path name, the `dc_unpack_design` utility cannot modify the file path name. For example,

```
'include "/u/design/include/package.h"
```

You must manually modify the path name in the RTL file.

